U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

PB-288 464

User Guide for the Enhanced Hydrodynamical-Numerical Model

Compass Systems, Inc, San Diego, CA

Prepared for

Corvallis Environmental Research Lab, OR

Jul 78

United States
Environmental Protection
Agency

Environmental Research
Laboratory
Corvallis OR 97330

EPA-600/3-78-074
July 1978

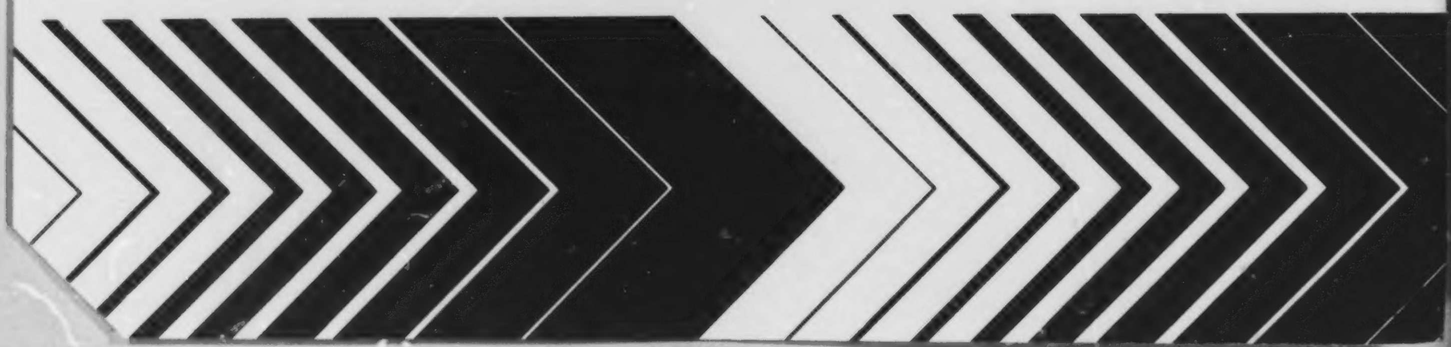
Research and Development

PB 288464

EPA

User Guide for the Hydrodynamical-Numerical Model

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161



RESEARCH REPORTING SERIES

Research reports of the Office of Research and Development, U.S. Environmental Protection Agency, have been grouped into nine series. These nine broad categories were established to facilitate further development and application of environmental technology. Elimination of traditional grouping was consciously planned to foster technology transfer and a maximum interface in related fields. The nine series are:

1. Environmental Health Effects Research
2. Environmental Protection Technology
3. Ecological Research
4. Environmental Monitoring
5. Socioeconomic Environmental Studies
6. Scientific and Technical Assessment Reports (STAR)
7. Interagency Energy-Environment Research and Development
8. "Special" Reports
9. Miscellaneous Reports

This report has been assigned to the ECOLOGICAL RESEARCH series. This series describes research on the effects of pollution on humans, plant and animal species, and materials. Problems are assessed for their long- and short-term influences. Investigations include formation, transport, and pathway studies to determine the fate of pollutants and their effects. This work provides the technical basis for setting standards to minimize undesirable changes in living organisms in the aquatic, terrestrial, and atmospheric environments.

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-600/3-78-074	2.	3. RECIPIENT'S ACCESSION NO. PB288464
4. TITLE AND SUBTITLE User Guide for the Enhanced Hydrodynamical-Numerical Model		5. REPORT DATE July 1978 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) A.D. Stroud and R.A. Bauer		8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Compass Systems, Inc. San Diego, California 92109		10. PROGRAM ELEMENT NO. 11. CONTRACT/GRANT NO. 68-03-2225
12. SPONSORING AGENCY NAME AND ADDRESS Environmental Research Laboratory-Corvallis Office of Research and Development U.S. Environmental Protection Agency Corvallis, Oregon		13. TYPE OF REPORT AND PERIOD COVERED Final 1975-1977 14. SPONSORING AGENCY CODE EPA/600/02

15. SUPPLEMENTARY NOTES

This is the companion document to EPA-600/3-78-073.

16. ABSTRACT

This guide provides the documentation required for used of the Enhanced Hydrodynamical-Numerical Model on operational problems. The enhanced model is a multilayer Hansen type model extended to handle near-shore processes by including:

- -Non-linear term extension to facilitate small-mesh studies of near-shore, including river inflow dynamics;
- -Layer disappearance extension to enable appropriate procedures in tidal flat and marshy regions, as well as some down/upwelling cases;
- -Thermal advection enhancement for treatment of thermal pollution cases by method of moments coupled with heat budget procedures;
- -Monte Carlo diffusion enhancement to deal with dispersion via statistical methods and comparison to the method of moments experiments.

The guide includes a description of the model system, source code maintenance procedures, notes on implementation, functional descriptions of routines and option sets, listings and example runs for a test data set.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS		b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
numerical methods	marshy regions		08/C
thermal pollution	operations research		12/A,B
monte carlo method			
mathematical methods			
near-shore			
river inflow			
tidal flat			

18. DISTRIBUTION STATEMENT Release to Public	19. SECURITY CLASS *(This Report)* Unclassified	21. NO. OF PAGES 152
	20. SECURITY CLASS *(This page)* Unclassified	22. PRICE PCA08 / MF A01

EPA Form 2220-1 (Rev. 4-77)

☆ U. S. GOVERNMENT PRINTING OFFICE: 1978—797-714/233 REGION 10

EPA-600/3-78-074
July 1978

USER GUIDE FOR THE ENHANCED HYDRODYNAMICAL-NUMERICAL MODEL

by

A. D. Stroud
and
R. A. Bauer
Compass Systems, Inc.
San Diego, California 92109

Contract Number 68-03-2225

Project Officer
R. J. Callaway
Marine and Freshwater Ecology Branch
Corvallis Environmental Research Laboratory
Corvallis, Oregon 97330

Corvallis Environmental Research Laboratory
Office of Research and Development
U. S. Environmental Protection Agency
Corvallis, Oregon 97330

DISCLAIMER

This report has been reviewed by the Corvallis Environmental Research Laboratory, U. S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the U. S. Environmental Protection Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

FOREWORD

Effective regulatory and enforcement actions by the Environmental Protection Agency would be virtually impossible without sound scientific data on pollutants and their impact on environmental stability and human health. Responsibility for building this data base has been assigned to EPA's Office of Research and Development and its 15 major field installations, one of which is the Corvallis Environmental Research Laboratory (CERL).

The primary mission of the Corvallis Laboratory is research on the effects of environmental pollutants on terrestrial, freshwater, and marine ecosystems; the behavior, effects and control of pollutants in lake systems; and the development of predictive models on the movement of pollutants in the biosphere.

A. F. Bartsch
Director, CERL

PREFACE

This Users Guide to the Optimized Hydrodynamical-Numerical (HN) Model system reported by Bauer and Stroud (1) details the information needed to use the model on CDC equipment. The reported efforts focused on a set of specific objectives and as a result the Users Guide and program library that evolved exclude the display program and other programs that are logically related to the HN model but that were not needed.

HN model work at Fleet Numerical Weather Central (FNWC), Monterey, was begun in 1966 by Dr. T. Laevastu and was continued at various levels of effort into this project. In 1971 the model work was shifted to the Naval Environmental Prediction Research Facility (NEPRF), when an Oceanographic Department was established under Dr. Laevastu. During the early years of the use of the HN model at FNWC and NEPRF, each new application resulted in the creation of a new program deck and often each new application also required features in the model that were unique to the problem under investigation. As a result by 1973 the number of separate models and special features became unmanageable in terms of both man time and computing resources and a major effort was undertaken to create a single program with selectable options that could be used on any geographic configuration. This effort resulted in the Optimized Model. Unfortunately the timing of this effort caused the programs related to the HN model to be distributed between the NEPRF CDC 3100, where quick turn around was possible, the Lawrence Berkeley Laboratory (LBL) CDC 7600, where cost effective calculations could be done, and FNWC's CDC 1604, where existing routines could be used without program conversion. The major elements of the optimized system referred to in Bauer (2) as Phases I, II and III were sent to Mr. Callaway at Environmental Protection Agency (EPA) Corvallis in 1975, where Phase I, a CDC 3100 program, is run on Oregon State's CDC 3300 computer and Phase II is run on the Bonneville Power Authority (BPA) CDC 6500. The original CDC 3100 program was converted to run on BPA's 6500 and extended to contour water heights by EPA. Since this program has concentrated on efforts of specific calculations rather than the description of a particular area, Phase III was not used and as a result is not included in this documentation.

Other major related programs that have not been included are the Varian contour plot, velocity vector plot program written for the CDC 3100 by J. Harding at NEPRF (3), the wave program written by Bauer (4) for the CDC 6500 that with minor modification could be used to generate the HN bathymetric data from contour charts, and finally the AT model sequence (Bauer et. al. [5]) that can be used to produce three hourly wind fields using the FNWC hemispheric fields to bound the local area grid solution forecasts using Bengtsson's limited area 3 parameter model.

The capabilities of the HN model have been considerably enhanced by the effort to develop and test specific modifications to the basic optimized model and can be considered operational. The goal of having a single documented model with many optional computations that can be used in any area has been realized. Additional effort is required to convert the related programs and merge them into the program library; however, with the disestablishment of the NEPRF Oceanographic Department and the reorganization of the EPA Corvallis Laboratory that occurred during the project, it is not clear what facilities and agencies would be involved in future efforts. What is clear is that the level of understanding and development of this model evolved only after many man years of trial and error, so that we strongly recommmend that anyone who has never used the model arrange to have help in setting up his first model runs. We also strongly recommend against converting the program to other computer hardware unless there are long term commitments to use the model, because the conversion would automatically lose the flexibility and structure of the present program library and because the conversion and debugging effort required would certainly be many times the cost of producing the required results with the present set of programs.

ABSTRACT

This guide provides the documentation required for use of the Enhanced Hydrodynamical-Numerical Model on operational problems. The enhanced model is a multilayer Hansen type model extended to handle near-shore processes by including:

- Non-linear term extension to facilitate small-mesh studies of near-shore, including river inflow dynamics;
- Layer disappearance extension to enable appropriate procedures in tidal flat and marshy regions, as well as some down/upwelling cases;
- Thermal advection enhancement for treatment of thermal pollution cases by method of moments coupled with heat budget procedures;
- Monte Carlo diffusion enhancement to deal with dispersion via statistical methods and comparison to the method of moments experiments.

The guide includes a description of the model system, source code maintenance procedures, notes on implementation, functional descriptions of routines and option sets, listings and example runs for a test data set.

This report was submitted in fulfillment of contract No. 68-03-2225 by Compass Systems, Inc. under the sponsorship of the U. S. Environmental Protection Agency. This work covers the period from June 1975 to March 1977 and was completed March 31, 1977.

CONTENTS

Foreword iii
Preface. iv
Abstract vii
Figures. x
Tables x
Acknowledgment xi

1. Introduction. 1
2. Recommendations and Conclusions 2
3. The Enhanced HN Model Program System. 4
    FORTRAN program structure. 4
    The library maintenance system 7
    Program library structure and usage. 9
4. HN Model Implementation Notes 17
    Implementation process 17
    Phase II timing notes. 20
    Notes on experimental/proposed features/methods. . .22

References 43

Appendices

A. Phase I Program Notes .45
    Description of Phase I program 45
    Listing for Phase I sample run 52
B. Phase II Program Notes. 68
    Description of program LAYER3S 68
    Listing for Phase II sample run. 77
C. Non-Linear Modifications. 93
    CONVCT and CONVCU series 93
    CONPRT, CONPRU and CONPRV series 94
    Listing for non-linear correction sets 95
D. Layer Disappearance Modifications 98
    FLAT, FLATA and FLATB series 98
    FLATIO and FLATIP series 101
    Listing for layer disappearance correction sets. . 103
E. Monte Carlo Modification
    MONTE. 110
    Listing for Monte Carlo correction sets. 112
F. Phase IV Program Series 116
    Thermal advection program. 116
    Listing for Phase IV sample run. 122

Preceding page blank

FIGURES

Number Page

1 The HN model system of programs 5
2 Intermediate data set format. 6
3 HN Phase I program structure. 10
4 HN Phase II basic program structure 12
5 HN Phase II enhancement impacts 13
6 Phase IV program structure. 15

TABLES

Number Page

1 Master Control Card Set 25
2 HN Phase I Control/Data Cards 28
3 UPDATE Directives Examples. 31
4 CDC UPDATE Control Card Parameter Examples. 31
5 HN Library: Phase I Modules 32
6 HN Library: Phase II Modules. 33
7 HN Library: Phase IV Modules. 36
8 HN Library Usage Job Control Deck Setup for CDC6500 38
9 HN Phase I Control/Data Card Set. 39
10 HN Master Control Card Set (All Phases) 40
11 HN Library Usage By CDC UPDATE Directives Set 41
12 HN Phase II Timing Performance Chart. 42

D-1 Tidal Flat Code Conventions 100

ACKNOWLEDGMENT

We wish to express our appreciation to R. Callaway, Project Officer, for providing the proper environment to perform theoretical studies of the model and wish to note the thoughtful user oriented feedback from C. Koblinski, also of the Corvallis laboratory.

We wish to thank Dr. T. Laevastu for his guidance and to acknowledge all of his innovative enhancements to the original Hansen model, which formed the basis for the optimized model.

We wish to thank the Naval Environmental Prediction Research Facility for allowing use of their facilities; S. Larson, K. Rabe and J. Harding for their many constructive suggestions; and the NEPRF computer support personnel for their assistance.

We wish to acknowledge the use of the ERDA computing facilities at Lawrence Berkeley Laboratory, University of California at Berkeley, California.

Finally, we wish to express our thanks to S. Haines for preparing the illustrations and tables and to D. Bauer for typing the report.

SECTION 1

INTRODUCTION

The Hansen-type optimized multi-layer Hydrodynamical-Numerical (HN) model is a numerical difference solution of the vertically integrated form of the hydrodynamic equations. Analytic details are presented in the enhanced model described by Bauer and Stroud (1). Included in this document are descriptions of the Data Initialization (Phase I), Hydrodynamic Computational (Phase II) and Auxiliary Computational (Phase IV) HN programs; the program library structure, usage and maintenance; and notes on implementation of the model. Program notes, listings, sample runs and module listings are included as appendices.

SECTION 2

RECOMMENDATIONS AND CONCLUSIONS

The HN Model program designed to permit options, boundary specification, grid size and number of layers to be specified by control cards is operational. The model consists of four separate programs, Phases I, II, III and IV. All are now coded to run on a CDC 6000-7000 series computer. A single program library documented in this Users Guide includes Phases I, II and IV. Phase I produces the initial geographic data set. Phase II, the main computational routine, solves the hydrodynamic equation set which can include as options the non-linear convective term and the treatment of areas where a layer disappears (tidal flat). As an option it also advects a set of points using Monte Carlo techniques. Phase IV uses the HN velocity fields as input to the Pedersen-Prahm advection method and decays the heat content of the advected volume.

Both Phase I and II have been tested on both synthetic data developed to test boundary and layer disappearance cases and on real data cases. Results on all cases attempted have been satisfactory. Results from Phase IV reflect the assumption of pure advection with no diffusion. In a real data test the plume developed by Phase IV was more compact and hotter downstream from the source than was observed by IR overflights.

From the experience gained using the HN model it is clear that valuable information can be obtained about the character of an area and that model results can aid both in the design of field experiments and in interpretation of field data. With the present model initial results for most areas can be obtained with one or two days of effort. While the thermal advection part of the model did not verify as well as hoped because of the assumption of no diffusion, this single example of the use of the HN derived velocity field should not discourage others who need to know spatial and temporal variation in the flow fields to design and interpret measurement or to derive input for other models.

SECTION 3

THE ENHANCED HN MODEL PROGRAM SYSTEM

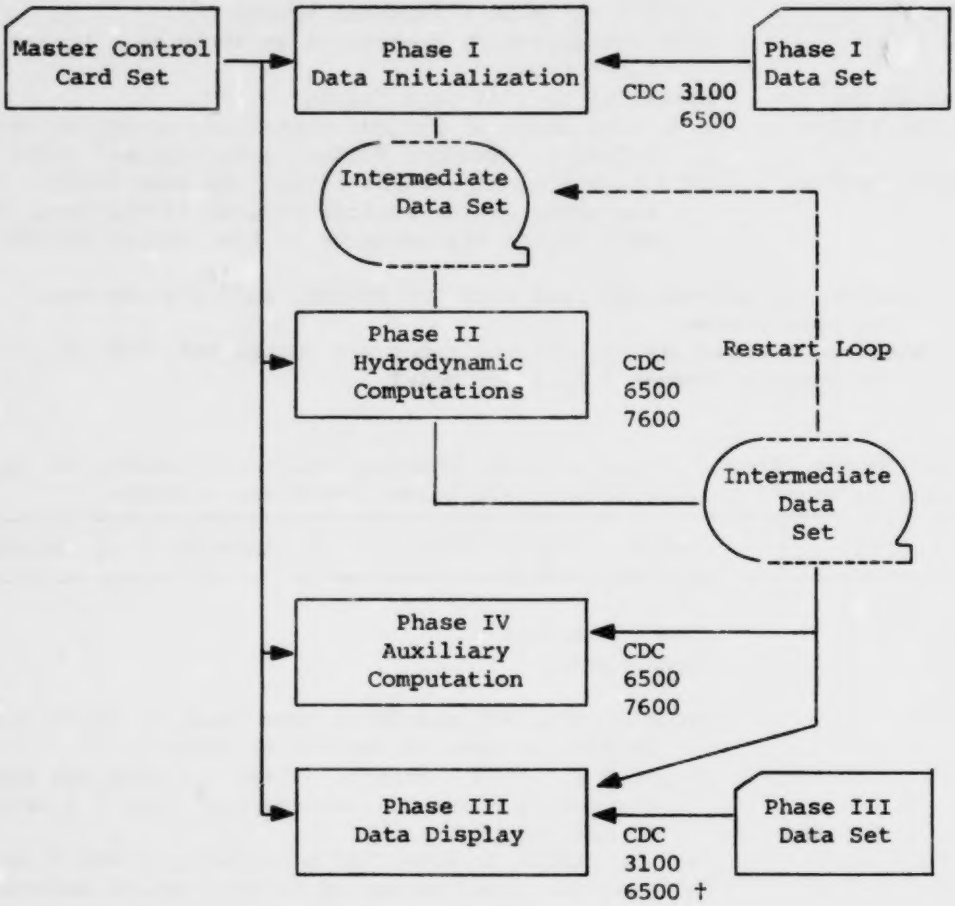
The enhanced HN model is composed of a series of FORTRAN programs maintained on a program library which is accessed through a library maintenance system. The structure of the program system, of the library system and of the resulting program library are presented. More detailed documentation of the individual programs, sample runs and enhancements are included as appendices.

FORTRAN PROGRAM STRUCTURE

The structure of the enhanced HN model as shown in Figure 1 is composed of four separate programs linked by the master control card and intermediate data sets. The four phases are: Phase I-Data Initialization; Phase II-Hydrodynamic Computations; Phase III-Data Display and Phase IV-Auxiliary Computations. The design of the sequence of programs allows the user to verify, save and reuse key intermediate results and minimizes the computer resources required during the computational phases.

The Phase I data initialization program is designed to check the master control card set defined in TABLE 1 and to create the intermediate data set with type 1 and type 2 time 0 records as defined by Figure 2 based on the water depths at U and V points and the Phase I data set, defined in TABLE 2.

The Phase II hydrodynamic computation program is usually the most expensive program to run in the sequence and, since it must be run out for at least a half tide cycle before the results become usable, one of the major goals in designing the system was to allow the Phase II program to be stopped


Master Control Card Set
Phase I Data Initialization
Phase I Data Set
CDC 3100 6500
Intermediate Data Set
Restart Loop
Phase II Hydrodynamic Computations
CDC 6500 7600
Intermediate Data Set
Phase IV Auxiliary Computation
CDC 6500 7600
Phase III Data Display
Phase III Data Set
CDC 3100 6500 †

† Conversion of Phase III from CDC 3100 to CDC 6500 was performed at EPA Corvallis. The CDC 3100 program is documented in (2).

Figure 1. The HN model system of programs.

ARRAY	A 2 dimensional matrix used in the HN model system with rows and columns indexed from the upper left hand corner
CHARACTER	6 bit CDC display code
BYTE	24 bits integer value
HEADER	First 5 bytes of a physical record
DATA FIELD	Byte containing an element of an array as a scaled ± integer value
PHYSICAL RECORD	Header +1 to 1245 data fields
LOGICAL RECORD	1 or more physical records containing an entire array stored in standard FORTRAN array element order
GROUPED RECORDS	2, 3 or 6 logical records having the same headers where the order of the logical records in the group is the only key to the contents of the logical record

Records are written and read with CDC FORTRAN Buffer statements in binary mode.

When the data set is on tape the tapes are unlabelled, 556 BPI, 7 track in stranger format 1 file per model.

Header format (first 5 bytes of every physical record.) Headers on the tape are in ascending sort order. All header bytes are + integers.

TIME	LAYER	TYPE	ROW-COL	MODEL NO

TIME	Time in seconds
LAYER	Layer number
TYPE	
1	Symbolic HTZ, HTU and HTV arrays (must be first record in set. Placed on the set by Phase I.
2	Computed ζ, U and V arrays. (Time 0 fields are placed on the set by Phase I. Records with time > 0 are placed on the set by Phase II.)
3	Used by FLATIP to save FLAT modified ζ, U and V arrays
4-7	Reserved for other groups of triple logical records
8	Double logical record group used to input variable winds from the AT model
9	Single logical record used by FLATIO to save character array for display, length=(MAXDIM+9)/10+2
10	Single logical records used by CONPRV to save convective terms (one separated set per IO step)
20	Hex grouped logical record used by Phase IV on a separate data set to save results and provide restart capability
ROW-COL	(Number of rows)*10000B+(number of columns)
MODEL NO	Number used to identify data set and checked against the MODEL number given in the master control card set

Figure 2. Intermediate data set format.

and restarted easily. This feature allows the initialization period results to be produced once and used to restart the model with different features activated in each restart. This is accomplished by having the Phase II program read the intermediate data set type 1 record to verify the correct data set is being read and to set up the topography, then to continue reading the intermediate data set until the type 2 record with a time matching the start time specified in the master control card deck is found.

The intermediate data set in conjunction with the master control card deck allows both Phases III and IV to produce products based on the HN computations after the hydrodynamic computational data rather than as an integral part of it, simplifying the logic and minimizing the memory and memory swapping required to obtain results.

The Phase III display program is not included in this document since it is highly dependent on the hardware available to the user. The Phase III program available to the authors uses the CDC 3100 and CalComp plot routines in a highly overlaid structure required to fit within the 26,000 24 bit words of the 3100 memory. This Phase III program has not been changed since it was reported by Bauer (2). The CDC 6500 version reprogrammed and extended by EPA at Corvallis would serve as a better program base for future efforts.

The auxiliary computations, Phase IV, use the velocity fields in the type 2 record from the intermediate data set to drive the duplex thermal advection model. The reasons for separating the auxiliary computations are that the advection processes can be run on longer time steps than the hydrodynamical model without producing noticeable differences in the results and the thermal advection model has its own relatively large memory requirements.

THE LIBRARY MAINTENANCE SYSTEM

This section describes the general structure of the CDC library maintenance system, UPDATE (6), as used for the HN programs. The HN program library is usually maintained on a magnetic tape or in permanent disk file accessed by UPDATE. UPDATE supports the following modules: decks, common decks and correction sets. Each module consists of a specific set of card images and is uniquely named. Each card image is identified by reference to the module name and a sequence number assigned to the card. As used with the HN program library, a deck usually consists of one or more routines. A common deck is a subset of card images that appears several times in the program but only once in the library. A simple reference to the common deck at the required location causes UPDATE to replace the reference card with the common deck cards referenced. This technique greatly reduces the effort required to maintain FORTRAN (7) COMMON blocks and other repeated sections of FORTRAN code.

Correction sets include library maintenance directives and associated new or replacement card images which modify existing modules. Correction features available include text insertion before, after or in place of specified card images, deactivation or reactivation of card images or prior correction sets and permanent structural modifications (renaming, renumbering, adding or purging modules). Most correction actions are fully recoverable since deactivated modules or card images are not removed from the library unless specifically directed during new library generation. Card images on the library are maintained in a compressed format which allows retention of current and historical status for each image and module.

After an initial library creation, use of the system merely requires that access be provided to the library and additional desired correction sets. Provided the appropriate access, the library system tailors the library based on the correction directives to form an expanded format subset of the library suitable for input to the CDC FORTRAN EXTENDED compiler, usually on file COMPILE. Use of the library system avoids the much less efficient process of maintaining and using actual card decks with large systems of programs. A more detailed and precise documentation of the CDC UPDATE library system features may be found in (6).

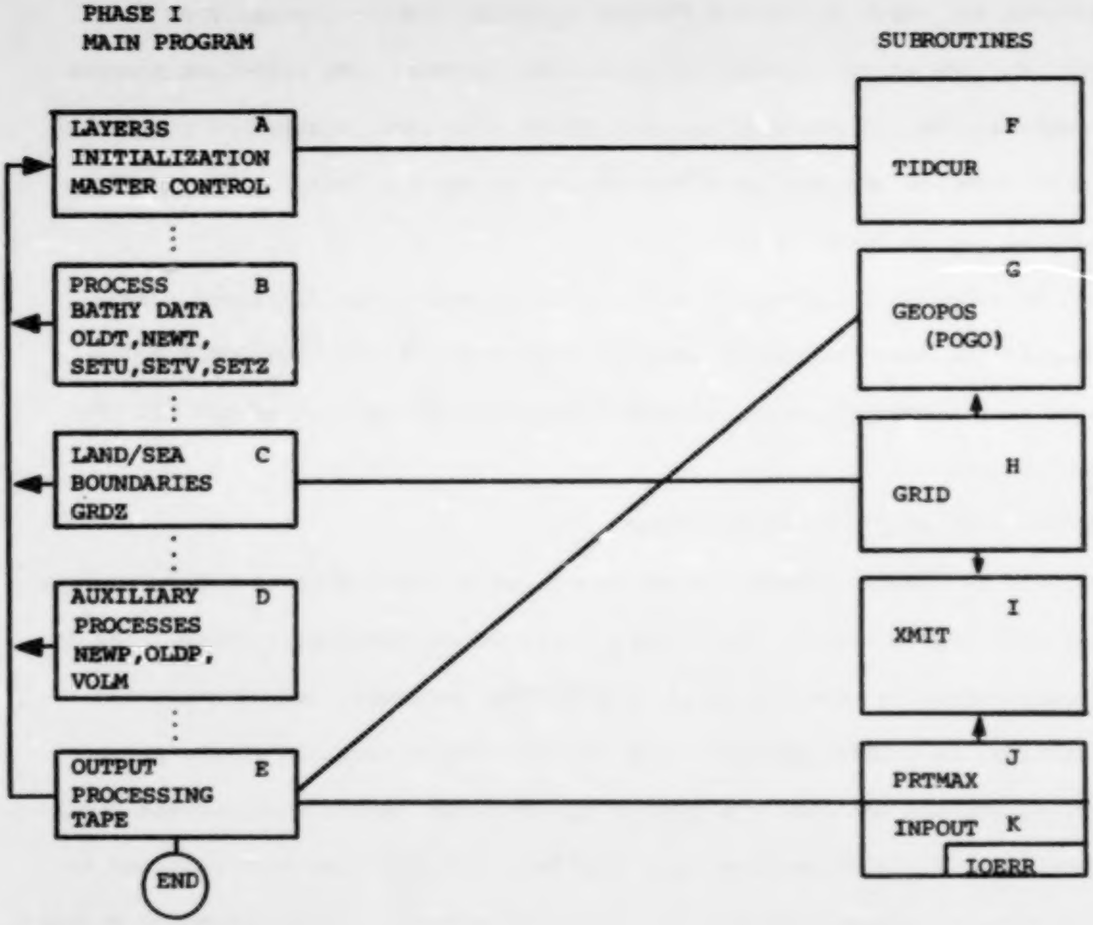
In order that correction sets may be presented and discussed, TABLE 3 presents the most frequently used CDC UPDATE directives, and TABLE 4 presents some frequently used parameters required for control of the CDC UPDATE program.

PROGRAM LIBRARY STRUCTURE AND USAGE

The HN program library is maintained on magnetic tape in a form compatible with UPDATE for CDC 6000/7000 et. al. series computer systems. The HN library currently consists of several FORTRAN programs, sub-programs and variations as UPDATE modules. Some of the modules retained in the present library version are the result of prior evolution rather than current development effort. Such modules, e.g. VARWIND, VARCORL, have been retained as they provide valuable options for future HN efforts. Other potential modules with similar or greater value have not been specially added due to separate evolution and lack of specific requirement. As the variety of modules present on the HN library involves overlapping deck requirements or multiple deck impact, the library has been maintained with "simplest" Phase II usage in mind.

Phase I

The library modules associated with Phase I have been presented in TABLE 5, including two decks and various correction sets. Figure 3 presents the resultant program structure of Phase I as a block diagram with references to the individual card image identification associated with each block. Appendix A


PHASE I
MAIN PROGRAM
SUBROUTINES
LAYER3S A
INITIALIZATION
MASTER CONTROL
PROCESS B
BATHY DATA
OLDT,NEWT,
SETU,SETV,SETZ
LAND/SEA C
BOUNDARIES
GRDZ
AUXILIARY D
PROCESSES
NEWP,OLDP,
VOLM
OUTPUT E
PROCESSING
TAPE
END
F
TIDCUR
G
GEOPOS
(POGO)
H
GRID
I
XMIT
J
PRTMAX
INPOUT K
IOERR

BLOCK KEY	DECK	MODULE ID LIMITS
A	HNI	HNI.2,HNI.54
B	HNI	HNI.55,HNI.76 ; HNI.85,HNI.91
C	HNI	HNI.77,HNI.84
D	HNI	HNI.122,HNI.143
E	HNI	HNI.92,HNI.121
F	HNI	HNI.237,HNI.292
G	HNI	HNI.194,HNI.236
H	HNI	HNI.144,HNI.193
I	UTIL	HNG5.28,UTIL.66
J	UTIL	HNG8.127,UTIL.98
K	UTIL	PR3.6,UTIL.63

Figure 3. HN Phase I program structure.

provides documentation of the Phase I program and associated subprograms including a FORTRAN source program listing with complete card image identification as provided by UPDATE on file COMPILE and printed output examples.

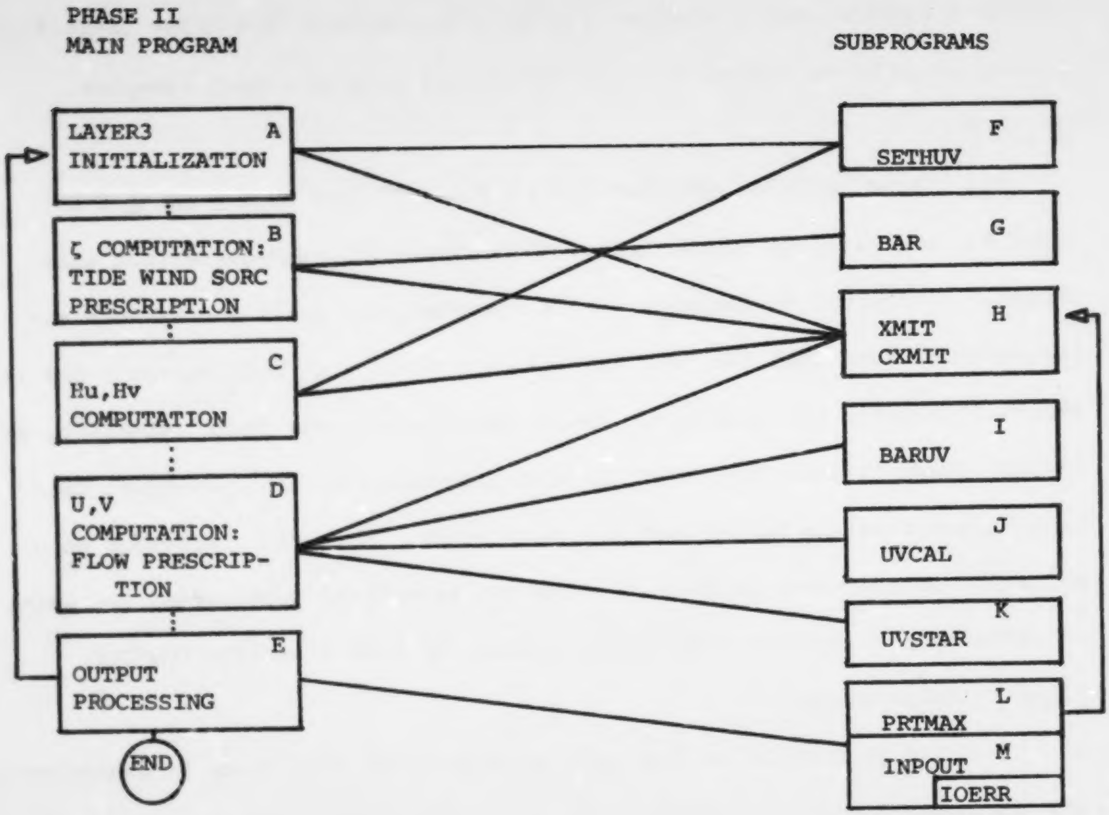
Phase II

The library modules associated with Phase II have been presented in TABLE 6, including two common decks, three decks and many correction sets showing the logical development of the computational model and various enhancements. Note that the deck UTIL is associated with both Phases I and II. Figure 4 presents the resulting program structure of the basic (simplest, enhanced) Phase II as a block diagram with references to the individual card image identification associated with each block. Appendix B provides documentation of the basic Phase II program and associated subprograms including a FORTRAN source program listing as provided by UPDATE on file COMPILE.

Phase II Enhancements

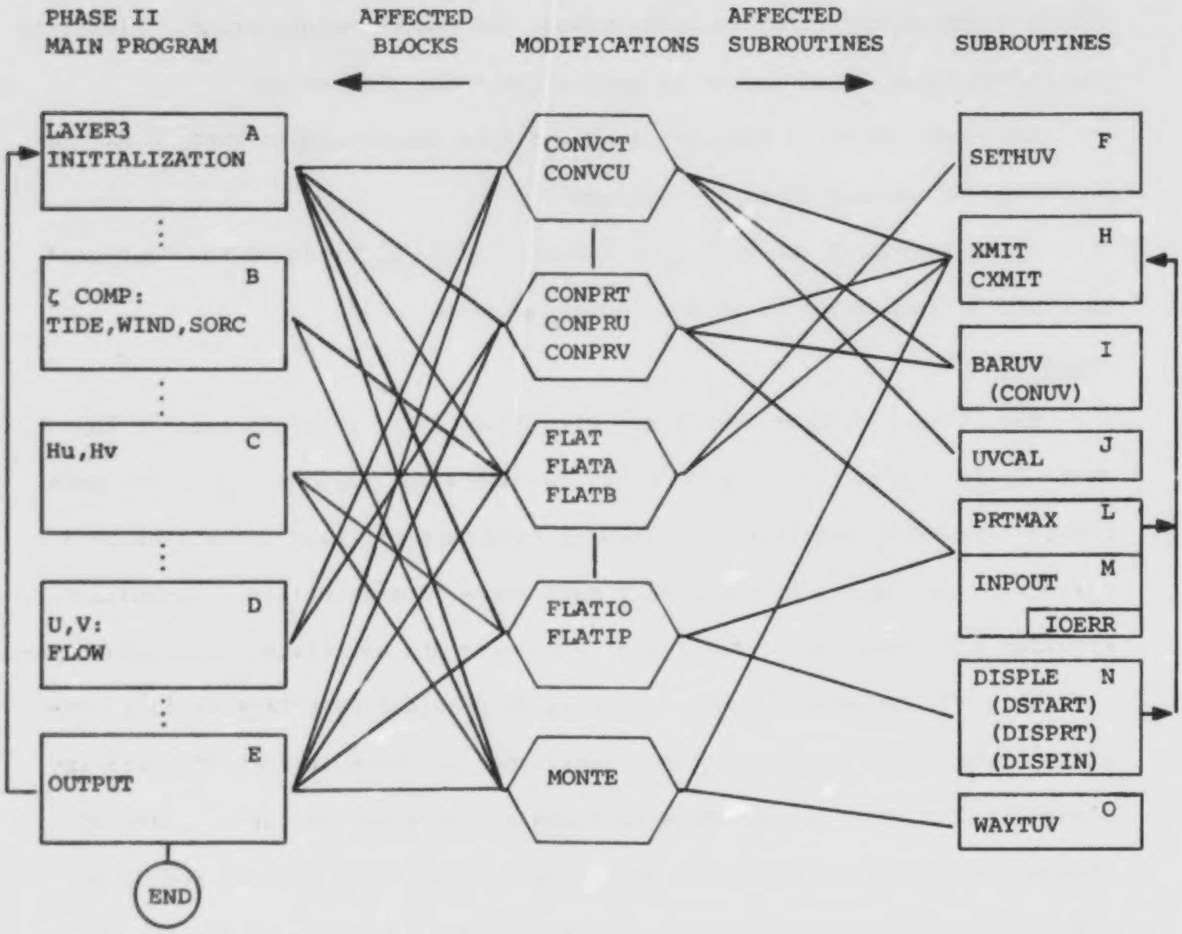
Included in TABLE 6 are the modules associated with Phase II enhancements and the module series associated with suppressing those enhancements until specific usage is required. Correction set names beginning with the letter A have been used to identify modules that exist merely to suppress features that are not a part of the basic Phase II program. If these suppression modules are inactivated through the UPDATE YANK directive or if specific subdirectives of such a module are inactivated by means of the UPDATE DELETE directive, the desired features becomes a part of the resulting FORTRAN source deck image on the COMPILE file.

Figure 5 presents the impact of each of the enhancement module groups on the Phase II program in a block diagram similar to Figures 3 and 4. The hexagonal blocks identify logical module groups that should not be separated for independent use. The one exception to this rule is that CONPRT may be


PHASE II
MAIN PROGRAM
SUBPROGRAMS
LAYER3 A
INITIALIZATION
B
ζ COMPUTATION:
TIDE WIND SORC
PRESCRIPTION
C
Hu,Hv
COMPUTATION
D
U,V
COMPUTATION:
FLOW PRESCRIP-
TION
E
OUTPUT
PROCESSING
END
F
SETHUV
G
BAR
H
XMIT
CXMIT
I
BARUV
J
UVCAL
K
UVSTAR
L
PRTMAX
M
INPOUT
IOERR

BLOCK KEY	DECK	MODULE ID LIMITS
A	HNG	HNG.2,HNG.78
B	HNG	HNG8.26,HNG.133
C	HNG	HNG.134,BC8.6
D	HNG	BARMOD.1,HNG8.48
E	HNG	HNG.191,HNG.246
F	HNG	HNG.249,HNG8.71
G	BARSTAR	HNG8.105,BARSTAR.36
H	UTIL	HNG5.28,BC8.16
I	BARSTAR	BARMOD.8,BARSTAR.50
J	HNG	HNG8.72,HNG8.94
K	BARSTAR	HNG8.95,BARSTAR.23
L	UTIL	HNG8.127,UTIL.98
M	UTIL	PR3.6,UTIL.63

Figure 4. HN Phase IV basic program structure.


PHASE II
MAIN PROGRAM
AFFECTED
BLOCKS
MODIFICATIONS
AFFECTED
SUBROUTINES
SUBROUTINES
LAYER3 A
INITIALIZATION
ζ COMP: B
TIDE,WIND,SORC
Hu,Hv C
U,V: D
FLOW
OUTPUT E
END
CONVCT
CONVCU
CONPRT
CONPRU
CONPRV
FLAT
FLATA
FLATB
FLATIO
FLATIP
MONTE
SETHUV F
XMIT H
CXMIT
BARUV I
(CONUV)
UVCAL J
PRTMAX L
INPOUT M
IOERR
DISPLE N
(DSTART)
(DISPRT)
(DISPIN)
WAYTUV O

BLOCK KEY	DECK	MODULE ID LIMITS
A-B	HNG	(SAME AS FIGURE 4)
C	HNG	FLAT.7,MONTE.61(or,FLAT.51)
D-E	HNG	(SAME AS FIGURE 4)
F	HNG	FLAT.52,HNG8.71
H-M	HNG.UTIL	(SAME AS FIGURE 4)
N	BARSTAR	FLATIO.4,FLATIO.66
O	BARSTAR	MONTE.64,MONTE.76

Figure 5. HN Phase II enhancement impacts.

used with CONVCT and CONVCU without CONPRU and CONPRV. Obviously the CONPRT module group should never be used without the CONVCT module group. Similarly the FLATIO group should never be used without the FLAT group.

Any combination of Features 1, 2 and 4 as identified in TABLE 6 may be used with or without their I/O options.

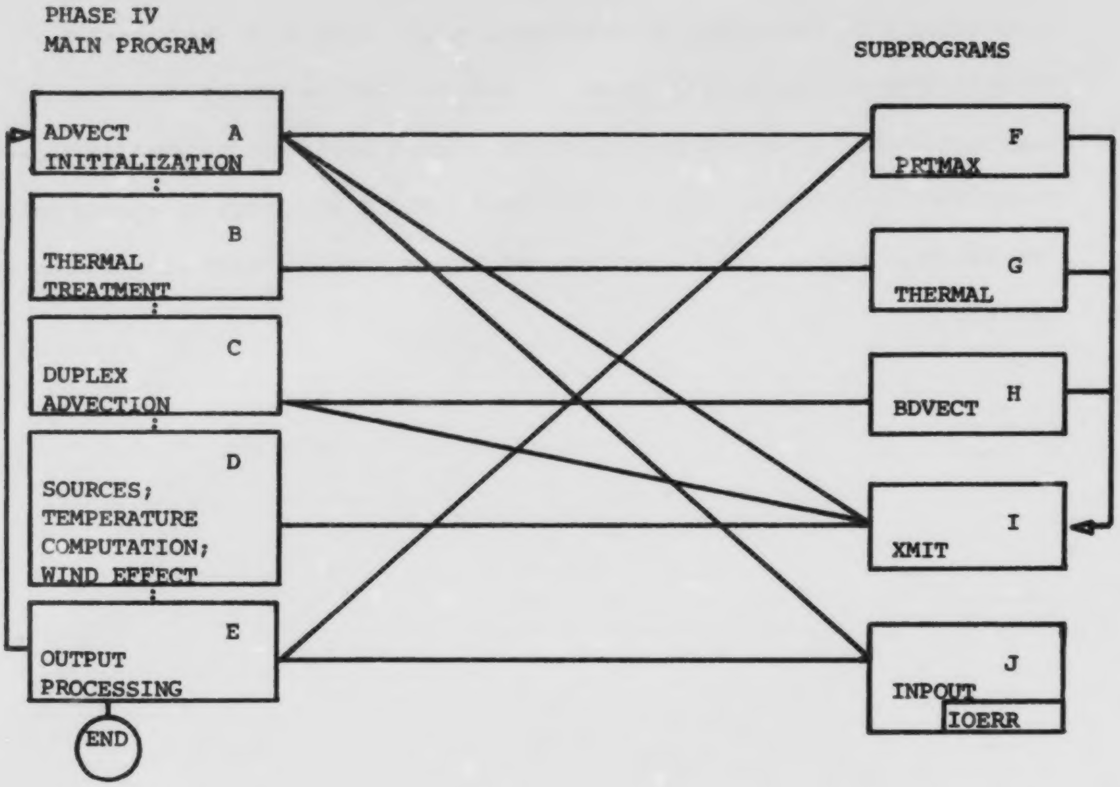
Appendices C, D and E provide library module listings of the Features 1, 2 and 4, respectively, with associated documentation.

Phase IV

The library modules associated with Phase IV have been presented in TABLE 7, including five common decks, two decks and various correction sets. Figure 6 presents the resultant program structure of Phase IV as a block diagram with references to the UPDATE card image identification. Appendix F provides documentation of the Phase IV program and associated subprograms including a FORTRAN source program listing as provided by UPDATE on file COMPILE. Note that routines with the same names as those in deck UTIL are included in deck AD. Though these routines are somewhat different Phase IV should ultimately evolve toward use of deck UTIL. That it does not do so currently is due to the separate initial library evolution of the thermal advection procedures.

Usage

TABLE 8 presents a basic CDC 6500 SCOPE (8) job control deck setup for use of Phases I, II and IV including a list of the various logical unit designations required by the job stream. TABLES 9 and 10 present the HN control/data card decks that are required for the particular HN phase desired in the deck setup of TABLE 8. TABLE 11 presents various UPDATE directive sets to achieve particular HN objectives. Both the YANK directive and the DELETE directive have been employed to remove the suppression directives


PHASE IV
MAIN PROGRAM
SUBPROGRAMS
ADVECT INITIALIZATION A
B THERMAL TREATMENT
C DUPLEX ADVECTION
D SOURCES; TEMPERATURE COMPUTATION; WIND EFFECT
E OUTPUT PROCESSING
END
F PRTMAX
G THERMAL
BDVECT H
I XMIT
J INPOUT
IOERR

BLOCK KEY	DECK	MODULE ID LIMITS
A	AD	A2.1,AD.39
B	AD	A3.5,A3.11
C	AD	A2.47,A2.52
D	AD	A2.53,A2.78
E	AD	A2.79,A1.27
F	AD	A1.28,A1.61
G	TH	T1.9,TH.324
H	AD	A2.97,A3.26
I	AD	A1.62,A1.65
J	AD	A1.81,A1.144

Figure 6. Phase IV program structure.

associated with correction modules whose names begin with the letter A. Note that to achieve a basic Phase II run, no YANK or DELETE directive is required. The "Q" option on the UPDATE control card must be used as shown in TABLE 8 in conjunction with the indicated COMPILE directive to access only the required decks. The correction set idents used in TABLE 11 examples are arbitrary.

SECTION 4

HN MODEL IMPLEMENTATION NOTES

This section discusses the process of implementing the HN model for a proposed application including the estimation of cost for the hydrodynamic computational phase. Also notes on various experimental or proposed features, methods and results are presented.

IMPLEMENTATION PROCESS

Throughout the implementation process consideration must be given to resources available for the application, including funds, computer resources, data resources and personnel. Success of the application will depend primarily on the quality and quantity of these resources.

Preliminary Processes

Site selection is an important step that interacts with available data resources. Features of interest for the study will play an important role in determining the number of layers as well as determining the extent of the area to be covered and the grid mesh length to be used. Interest should interact with practicality here to ensure economic implementation. Bathymetric features, tidal character, storm surge and river inflow all need to be considered in determining how best to cover the area of interest.

Orientation of the grid with respect to land is not material in itself; however, economical use of computer memory would point toward aligning at least one side of the grid with the coastline. Inclusion of river estuaries is preferable to attempting to prescribe river inflow across the estuary mouth. Some additional notes on river inflow methods may be found in the proposed methods section. Note that specific area selection determines the high tide maximum depth, and selection of the grid mesh size determines the maximum practical time step through the Courant-Fredricks-Lewin (CFL) stability criterion:

$$\ell/\tau \geq (2gHmax)^{\frac{1}{2}}.$$

Once the time step is known an estimate of cost of Phase II on the CDC 7600 at Berkeley can be made by referring to TABLE 12 and the notes on Phase II timing.

The actual grid preparation is straightforward once the approximate area has been outlined on the appropriate chart and the grid size has been selected. Merely rule the z grid carefully on the chart; then read depth values from the chart row by row for the hu and hv fields respectively. The first hu value is taken one-half grid point to the right of the z(1,1) point and thus along each row for ME values. When a u or v point overlies land, a zero value is used. During this digitizing process it may be important to shift or rotate features slightly in order to produce realistic cross sections in the depth fields. For this reason hand digitization is the most efficient method of preparing the depth fields; however, programs do exist which could be used to automate this process (4).

Phase I

When the hu and hv depth decks are prepared along with the necessary Phase I control cards (TABLES 8, 9 and 11) and the data are gathered to prepare the master control card set (TABLES 1 and 10), Phase I can be run to verify the depths and control card data.

Section 3 defines the data requirement for the various HN model phases. TABLE 9 shows the data card sequence for these Phase I control cards and the depth data. TABLE 8 contains the computer job control card deck setup for Phase I and TABLE 11 provides the required library directive set to access the Phase I program on the HN program library.

Phase I provides various printed output options to assist the implementation. All Phase I control and data cards are verified as to format and printed. A printed plot is presented for each TIDE card present with an indication of appropriate offset to initialize properly the tidal input to Phase II. A printed symbolic array of the land-sea boundaries and set of error messages indicate required correction to the depth fields in order to define valid land-sea boundaries. Correction can then be made in subsequent runs by either using change cards or by repunching the depth cards. The CFL derived time step is printed for both the maximum depth specified on the LAYS card and the actual maximum depth in the field as verification of that computed during the grid preparation phase. The VOLM card allows computation of regional or section volumes. The TAPE card, in addition to causing the depth values to be written on the intermediate data set, provides various print options for the symbolic height and depth fields and reinitializes the Phase I program so additional Phase I control/data card sets can be processed in the run.

Phase II

The intermediate data set prepared by Phase I and the master control card set are used to execute Phase II. TABLE 8 shows a Phase II computer job control deck setup. TABLE 10 presents the HN Phase II control card set which provides access to and verification of the intermediate data tape during Phase II initialization. TABLE 11 provides the essentials for determining the appropriate library directives set to access the Phase II program on the HN program library.

Phase III

Use of HN Phase III on any computer system other than the CDC 3100 would require a conversion effort to the local plotting capabilities as well as to the local computer system. The documentation (2) of data requirements should prove adequate for most applications.

Phase IV

In its present form Phase IV is a thermal advection model using the method of moments based on Pederson and Prahm (9). With some additional effort to remove the thermal calculations the model could be used for the advection of other properties. TABLE 11 presents the required library directives set required to access the thermal advection package on the HN program library. TABLE 8 provides the computer job control deck setup for the thermal advection, and TABLE 10 presents the required HN control cards used to control and access the intermediate data tape for thermal advection implementation. The variable ITS on the TIME card should be set to the earliest start time of any source.

PHASE II TIMING NOTES

TABLE 12 presents timing information in the form of a "performance index." The HN performance index, I_{HN}, is a comparative measure of job performance for HN Phase II. The index is basically the length of time in CDC 7600 central processing unit (CPU) seconds required to perform all type computations for a single grid node and computational step. The total number of CDC 7600 CPU seconds required to accomplish a particular Phase II objective can be estimated by using a composite index to describe the job. For example, a composite index for a non-linear HN version including tidal flat capability and tidal flat I/O without saving any other data, including convective data, would be formed by using references 25, 23, 16 from the table $(1.56+.37+.16=2.09(X10^{-4}))$.

The index is then multiplied by the total number of sea points for all layers and the number of computational steps for the job to compute the CDC 7600 CPU second estimate.

Each of the indices for references B though 10 of TABLE 12 was computed directly from the accounting statistics for HN Phase II jobs as described. References 0-10 included in excess of 1000 computational steps for 260 sea points. References B and D included 567 sea points computed for in excess of 8000 computational steps. The number of sea points for a particular area may be computed as the summation of the symbolic height array from Phase I, as:

$$Ps = \sum_{I=1}^{NO} HTZ(I).$$

The number of computational steps may be computed readily from information on the Phase II TIME control card for the job, as:

$$Sc = (ITE-ITS)/ITINC.$$

Currently estimated CDC 7600 job cost at LBL Berkeley for about 1000 steps, 1000 sea points, an index of $2.0X10^{-4}$ and approximate cost per CPU second equal to 23¢ would be:

$$COST = .23P_s S_c I_{HN} = \$46.00.$$

The cost estimate will generally be conservative in the sense that it will overestimate the cost for long jobs but may underestimate for some very short jobs. A recent job with 3125 steps, 716 sea points and an estimated composite index of $2.25X10^{-4}$ resulting in a 505 CPU second estimate actually used 334 CPU seconds with resulting actual index of 1.50. The index discrepancy resulted primarily from the low tidal flat involvement relative to the index base.

A set of such indices for the particular computer system used are readily derived based on several actual jobs. Several of the component indices are extremely area specific, e.g., the tidal flat computations should vary extensively from area to area based on actual tidal flat involvement. Less than 25 % of the sea points in the tidal flat grid ever become involved in the tidal flat computations directly and even then for less than 50% of the model time. Note that the tidal flat I/O depends markedly on whether other data are saved during the job (MODOUT on TIME card $\neq \infty$.) This is due primarily to the requirement to reread the entire output data file in order to print the tidal flat change steps.

NOTES ON EXPERIMENTAL/PROPOSED FEATURES/METHODS

The following set of notes is intended to encourage additional thought and work on the HN model and its application to various areas. The existence of a successful model need not imply that the model is static. Much work needs to be done in areas that have been little explored, if at all, numerically or analytically.

Friction Term Modification

Experiments were performed which showed that replacing the smoothed with unsmoothed velocity elements of the friction term as indicated by the form of the differential equations in (1) was insignificant for tests performed on the tidal flat grid. After three hours of model time the maximum differences between comparable runs with the smoothed terms in the friction computation and with old velocities in the friction computation were $8.\text{X}10^{-3}$ cm for water heights and $2.8\text{X}10^{-2}$ cm/sec for velocity components.

Removal of a constraint restricting the magnitude of the friction computation to be no larger in absolute value than the magnitude of the smoothed velocity component from the prior step also proved insignificant. A difference between comparable runs with and without the constraint produced zero differences to the precision of the data saved on tape (10^{-3} cm, 10^{-3} cm/sec) after three hours of model time for the tidal flat area.

Actual computations for the various enhancement tests retained both the smoothed velocity term and the friction constraint to avoid introducing minor differences between most recent and earlier runs. Since both modifications were shown insignificant numerically, no major impact need be expected from inclusion of these changes in library correction set HNG9 for future usage. Further tests could be performed by using YANK feature on HNG9 set.

Notes on Modeling River Inflow

Extensive testing of river inflow specification has not been accomplished. If the river mouth lies on an open boundary without tidal specification, the boundary acts as an aperture to an infinite capacity channel. The problem is alleviated if actual river mouth heights can be specified. If the tidal values are not available, perhaps the most practical solution is to include a portion of the estuary as part of the grid specification leaving no opening for river inflow (a dead end channel) and then to specify reasonable values for volume on a FLOW card at the end-point of the channel (the last z point.) The channel should be least one Z cell in length to achieve the desired flow effect into the adjacent water mass. If the estuary volume and friction forces are modeled, tidal flows at the mouth should be reasonable even if the estuary is realigned in the grid to prevent an excessively large grid size or land-sea ratio.

Notes on Introducing Diffusion into Advective Schemes

Vertical diffusion could be readily incorporated into the existing model of moments by determining a volume increment to be applied at each step to all affected cells. Extremely arbitrary experiments were run testing the feasibility by using an increment that was proportional to the absolute value of the base ten logarithm of the total volume within the cell. The factor chosen was too large for a first guess and the volumes increased too rapidly. The factor probably should be based on the time step and the affected area within each cell.

Horizontal diffusion effects would require a complete rewrite of the method of moments advection scheme if they were to be incorporated therein.

TABLE 1. MASTER CONTROL CARD SET (page 1 of 3)

Card Type	Program Variable	Description
NAME Card		Format (20A4)
1-4		NAME
5-80		Comments or titles (any number of cards may be used to identify the output)
GRID Card		Format (A4,2I3,7E10.3)
1-4		GRID
5-7	NE	Number of rows
8-10	ME	Number of columns
11-20	DL	Distance between grid points (cm)
21-30	ROTANG	Counter-clockwise angles between north and the +Y axis of the computational grid (deg)
31-40	C	Wind drag coefficient
41-50	TK	Mid latitude of grid (deg)
51-60	R	Bottom friction coefficient
TIME Card		Format (A4,I6,7I10)
1-4		TIME
5-10	MODOUT	Number of seconds between saved results
11-20	ITO	Time of first output in seconds
21-30	ITS	Start time for computations in seconds. If this time is greater than zero, the intermediate tape is assumed to contain U,V and Z arrays for necessary levels at the time specified.
31-40	ITE	End time for computations in seconds
41-50	ITINC	Time step in seconds
51-60	IPO	Time of first printout
61-70	IPMOD	Number of seconds between printouts
FACT Card		Format (A4,6X,7E10.3)
1-4		FACT (when present must precede TIDE cards in Phase I)
5-10	Blank	
11-20	TIDSPD(1)	Speed of the tidal constituents
21-30	TIDSPD(2)	
31-40	TIDSPD(3)	(Match order of component phase angle and amplitude on TIDE card) K1 = 15.0410686 (deg/hr) O1 = 13.9430356 M2 = 28.9841042 S2 = 30.0000000 N2 = 28.4397295

TABLE 1 (Page 2 of 3)

Card Type	Program Variable	Description
TIDE Card		Format (A4,4I3,I6,4F7.2,4F6.2,2F3.2)
1-4		TIDE (0≤J≤4)
5-7	ITIDE(1,J)	Minimum row (N) index for tidal input
8-10	ITIDE(2,J)	Maximum row (N) index for tidal input
11-13	ITIDE(3,J)	Minimum column (M) index for tidal input
14-16	ITIDE(4,J)	Maximum column (M) index for tidal input
17-22	ITIDE(5,J)	Time offset between time of computations and time 0 tide phase angle (sec)
23-29	TIDE(1,J)	Tidal constituents 1 phase angle (deg)
30-36	TIDE(2,J)	2
37-43	TIDE(3,J)	3
44-50	TIDE(4,J)	4
51-56	TIDE(5,J)	Tidal constiuents 1 amplitude (cm)
57-62	TIDE(6,J)	2
63-68	TIDE(7,J)	3
69-74	TIDE(8,J)	4
75-77	TIDE(9,J)	Weight factor for second layer tide heights (no tide set in second layer if 0)
78-80	TIDE(10,J)	Weight factor for third layer tide heights (no tide set in third layer if 0)
WIND Card		Format (A4,4I3,4X4I10)
1-4		WIND (0≤J≤4)
5-7	IWIND(1,J)	Minimum row (N) index for wind field
8-10	IWIND(2,J)	Maximum row (N) index for wind field
11-13	IWIND(3,J)	Minimum column (M) index for wind field
14-16	IWIND(4,J)	Maximum column (M) index for wind field
17-20	Blank	
21-30	IWIND(5,J)	Wind start time (sec)
31-40	IWIND(6,J)	Wind end time (sec)
41-50	IWIND(7,J)	Wind direction (degs true from which wind blows)
51-60	IWIND(8,J)	Wind speed (m/sec) Note: This is the only measurement in the input given in meters rather than cm.
LAYS Card		Format (A4,I6,3F10.0,8F5.0)
1-4		LAYS
5-10		Number of layers
11-20	HL(1)	Maximum depth of layer 1
21-30	HL(2)	2 if zero the layer is
31-40	HL(3)	3 ignored in the computations

TABLE 1 (Page 3 of 3)

Card Type	Program Variable	Description
41-45	ALPHA(1)	Smoothing parameters for layer 1
46-50	ALPHA(2)	2
51-55	ALPHA(3)	3
56-60	RHO(1)	Density for layer 1
61-65	RHO(2)	2
66-70	RHO(3)	3
71-75	HMIN	Minimum allowable thickness of layer in Phase II
SORC Card		Format (A4,4I3,2I2,2I10,4E10.3)
1-4		SORC ($0 \le J \le 10$)
5-7	ISORC(1,J)	Minimum row (N) index for source
8-10	ISORC(2,J)	Maximum row (N) index for source
11-13	ISORC(3,J)	Minimum column (M) index for source
14-16	ISORC(4,J)	Maximum column (M) index for source
17-18	ISORC(5,J)	Minimum layer (LM) index for source
19-20	ISORC(6,J)	Maximum layer (LP) index for source
21-30	ISORC(7,J)	Start time for source
31-40	ISORC(8,J)	End time for source
41-50	SORC(1,J)	Concentration introduced per time step
51-60	SORC(2,J)	Volume of water introduced per time step per level (cm^3)
FLOW Card		Format (A4,4I3,2I2,2I10,4E10.3)
1-4		FLOW ($0 \le J \le 4$)
5-40	IFLOW(,J)	Same as SORC card
41-50	FLOW(1,J)	Direction of flow (deg true)
51-60	FLOW(2,J)	Velocity increment (cm/sec/step)
61-70	FLOW(3,J)	Second layer factor
71-80	FLOW(4,J)	Third layer factor
COMP Card		Format (A4,I6)
1-4		COMP This card is used to start the computations in the Phase II program. It is the last card read in the control card deck by the program Phase II.
5-10	MODEL	Model number used in record header on intermedate data set

TABLE 2. HN PHASE I CONTROL/DATA CARDS (page 1 of 3)

Card Type	Program Variable	Description
NEWT Card		Format (A4,F6.3,F10.0)
1-4		NEWT Read in new format HTU and HTV tables that follow card
5-10	FAC	Conversion factor used to change depth units to cm. If blank assumed = 1.
11-20	SEALEV	Sea level adjustment to be added to water depths in cm. (No adjustment or conversion occurs until TAPE card processing).
OLDT Card		Format (A4,F6.3,F10.0)
1-4		OLDT Read in old format HTZ, HTU and HTV tables that follow card
5-10	FAC	Conversion factor used to change depth units to cm. If blank assumed = 1.
11-20	SEALEV	Sea level adjustment to be added to water depths in cm. (No adjustment or conversion occurs until TAPE card processing.)
New Format HTU and HTV Tables		Format (12I6)
1-6 . . .	IU or IV	Follows NEWT card. Each row of the array is read with a separate READ statement. The array element (1,1) is punched on the first field of the first card and the last element in the row (1,ME) is the last value on the (ME-1)/12+1 card. The elements (2,1)...(NE,1) all begin new card sets. The IU array is read first followed by the IV array.
Old Format HTZ, HTU and HTV Tables		Format (12I6)
1-6 . . .	IZ, IU or IV	Follows OLDT card. Each array is read using logic that duplicates an array READ using a double indexed implied DO loop with the column index varying faster. Only the last card in each array set can contain unused fields. HTZ array is read first followed by the HTU and HTV arrays.
SETU Card		Format (A4,4I3,4X,I10)
1-4		SETU Set the value ICC in HTU within the limits

TABLE 2 (page 2 of 3)

5-7		Minimum row (N) dimension
8-10		Maximum row (N) dimension; if zero set to min value
11-13		Minimum column(M) dimension
14-16		Maximum column (M) dimension; if zero set to min value
17-20		Blank
21-30	ICC	Value to be placed in array (same units as HTU field prior to conversion)
SETV Card		Same as SETU. Modify HTV array.
SETZ Card		Same as SETV. Modify HTZ array.
VOLM Card		Format (A4,4I3,4X,I10)
1-4		VOLM
5-20		Same as SETU card
21-30		Reset flag 1 = volume to zero before computing volume 0 = add computed volume to previous volume
GRDZ Card		Format (A4,6XI10)
1-4		GRDZ Set HTZ and -1 values in HTU and HTV field; test boundary configuration; provide print option
5-10		Blank
11-20		Flag to bypass combined symbolic HTZ, HTU and HTV printout 0 = print 1 = bypass printout
OLDP Card		Format (A4,4I3)
1-4		OLDP Punch old format HTZ, HTU and HTV cards
5-16		Same as SETU card, except if left blank limits set to 1, ME or NE as appropriate
NEWP Card		Format (A4,4I3)
1-4		NEWP Punch new format HTU and HTV table
5-16		Same as OLDP card
TAPE Card		Format (A4,I6,I10)

TABLE 2 (page 3 of 3)

1-4		TAPE 1) Combine u,v depth print 2) Convert and adjust sea level as required 3) Print HTZ, HTU and HTV 4) Set HTZ, HTU and HTV values less than 0 to 0 and scale by 10 5) Write HTZ, HTU and HTV fields on tape 6) Set Z, U and V fields to 2 for all cells over water, -0 for all cells over land. Write on tape for layer 1. 7) Repeat 6 for layer 2 and then 3 if HL(2) and HL(3) are non-zero 8) Write double EOF on tape, backspace 1 EOF 9) Return to reset for next data set
5-10	ICC	Skip parameter >0 Print HTZ, HTU and HTV but do not write tape 0 Print and write tape
11-20	ICD	>0 Request U/V grid print of depth field ≤0 No print

TABLE 3. UPDATE DIRECTIVES EXAMPLES

DIRECTIVE	DESCRIPTION
*DECK A	The first card of deck A
*COMDECK B	The first card of common deck B
*IDENT C	The first card of correction set C
*DELETE A.3,A.7	Deactivates 3rd through 7th cards of deck A and substitutes following non-update text after card A.7
*BEFORE A.8	Inserts following text immediately prior to card A.8
*INSERT A.8	Inserts following text immediately after card A.8
*RESTORE A.5,A.7	Reactivates cards A.5 through A.7 and inserts following text after card A.7
*CALL B	Replaced by active cards in common deck B
*COPY A,A.9,A.11	Copies cards A.9 through A.11 of deck A into current correction set
*YANK C.D	Deactivate correction sets C through D
*ADDFILE	Insert following decks and common decks at the end of the last deck on new program library
*COMPILE A	Specifies decks to be prepared for compilation on file COMPILE

TABLE 4. CDC UPDATE CONTROL CARD PARAMETER EXAMPLES

CONTROL PARAMETER	FUNCTIONAL DESCRIPTION
C	Allows specification of compilation file name. The default name is COMPILE.
D	Allows 80 columns to be used for data sets, otherwise 72 columns
F	Full upate (used here only for creation of new library). All decks are processed.
I	Allows specification of an alternate input file name. The default name is INPUT.
K	Provides decks to compile file in sequence specified on the compile directive
L	List options for printed output
N	New program library . The default name is NEWPL
O	Allows specification of an alternate printed output file name. The default name is OUTPUT.
P	Old program library. The default name is OLDPL
Q	Quick update. Processes only decks listed on compile directives.

TABLE 5. HN LIBRARY: PHASE I MODULES

TYPE/ NAME	STATUS	DEPEN- DENCY	INCOMPAT- IBILITY	FEATURE	DESCRIPTION COMMENTS
DECK/					
UTIL	A				Includes routines:INPOUT, IOERR,XMIT,PRTMAX
HNI	A				Includes routines LAYER3S, GRID,GEOPOS,TIDCUR
CORREC- TION SET/					
HNJ	I	PR3 BC8	Phase II		Phase II to Phase I conversion of deck UTIL and 3100-6500 conversion of deck HNI. Inactive due to incompatibility.
BKYPRT	I	BKY FACILITY	Other sites		Required for correct overprinting at BKY for land/ sea boundary print feature in routine GEOPOS.
AHNIJ1	A				Inactivates HNJ, BKYPRT through YANK feature to make library Phase II compatible and 6500 site compatible.
HNK	A				Corrects indexing error in VOLM control card supporting code.
HNL	A				Corrects page break suppression so it works on 6500 and 7600.
PHASE1	E				Reactivates HNJ for non-BKY site use of Phase I code.

Status Code: A-Active on the library
I-Inactive on the library
E-Extra-library correction set

TABLE 6. HN LIBRARY: PHASE II MODULES (Page 1 of 3)

TYPE/ NAME	STATUS	DEPENDENCY	INCOMPATIBILITY	FEATURE	DESCRIPTION COMMENTS
COMDECK/					
CONTROL	A			B	COMMON/CONTROL/(includes 500 positions for local routine use)
BLANK	A			B	COMMON//(the major array variables for all layers)
DECK/					
HNG	A			B	Includes routines LAYER3, SETHUV,UVCAL
BARSTAR	A			B	Includes routines UVSTAR, BAR,BARUV
UTIL	A			B	Includes routines INPOUT, IOERR,XMIT,CXMIT,PRTMAX
CORRECTION SET/					
HNG4	A			B	HNG4 through HNG6 are some of the earliest corrections to the initial creation of the HN library. Earlier correction sets have been irrecoverably incorporated by resequencing decks.
HNG5	A	HNG4		B	
HNG6	A	HNG5		B	
HNG7	I	HNG6	HNG8		Minor corrections to the original optimized code superceded by HNG8.
HNG8	A	HNG6	HNG7	B	Major structural changes in optimized code toward complete layer generality. Introduced UVCAL, affects all decks.
VARWIND	I	HNG8	UTILA, PR3, CONPRU	W	Allows wind array input as Type 8 data records from tape logical unit 2. Requires review before use.
VARWINE	I	VARWIND		W	Revision of VARWIND for compatibility with more recent enhancements. Corrects rotation error.
VARCORL	I	HNG8	HNG9	C	Allows variable coriolis parameter for arbitarily rotated grid. Requires review prior to use.
HNG81	A	HNG8		B	Corrects rotation error in basic WIND feature, produces z centered velocity print, and Hu/Hv correction for layers 2+.

TABLE 6 (Page 2 of 3)

BC8	A	HNG8	HNJ	B	Uniform flow boundary condition to match that on first row and column
BARMOD	A	HNH8		B	Changes boundary conditions on 2nd order computations (BAR, BARUV.) Land/sea amplitude damping: ζ↓,U/V↑
PR3	A	HNG8	HNJ	B	Precision feature: sets precision of data on output tape to 10^{-3} instead of 10^{-1} cm (cm/sec)
UTILA	A		FLATIO VARWIND	B	Data type processing revision to INPOUT supporting CONPRU modification to IOERR for 6500 compatibility
UTILB	A	UTILA		B	Data type processing revision to INPOUT supporting CONPRV and VARWIND
LIB2	A			B	Corrects minor coding errors
HNG9	A		VARCORL	B	Corrects thickness check for bottommost layer; removes friction constraint; reverts from bar term usage to old velocity in friction term; removes non-linear interpolation scheme from star term
VALDEZ	I	CONTRM BLANKA		V	Changes single layer version for array storage
SINGLE	A	HNG5		B	Removes arrays for 2nd and 3rd layer; leaves single layer version. For array storage: MAXDIM=800
CONTRM	A	SINGLE BLANKA		B	Changes single layer version such that CONTROL arrays dimensioned 300
BLANKA	A	SINGLE		B	Changes single layer version for array storage: MAXDIM=300
CONVCT	I	BARMOD		1	Feature 1: non-linear term computations. Forms and uses ENTRY CONUV to routine BARUV for the procedure
CONVCU	I	CONVCT		1	Corrects prior version boundary condition to consistent boundary treatment normal vs. tangential
CONPRT	I	CONVCT		1	Feature 1 OUTPUT option: nonlinear terms printed in array form using PRTMAX (2 arrays)

TABLE 6 (Page 3 of 3)

CONPRU	I	UTILA CONPRT		1	Modify print option replacing printed arrays with data records TYPE8 on LU1	
CONPRV	I	UTILB CONPRU		1	Modify data save option replacing TYPE8 with TYPE10 data records for VARWIND compatibility	
FLAT	I	HNG8.1 BC8		2	Feature 2: layer disappearance enhancement follows thickness comp., allows tidal flat phenomena	
FLATA	I	FLAT		2	Corrects detail feature of recomputation flag field to assure proper tidal flat handling	
FLATIO	I	FLAT	UTILA	2	Feature 2 output option: tidal flat flag fields saved on LU1; printed at normal end of job, ends EOF LU1.	
FLATIP	I	FLATIO		2	Modifies FLATIO to simplify INPOUT processing in conjunction with UTILA	
MONTE	I			4	Feature 4: Monte Carlo enhancement follows thickness comp. (FLAT comp. if present) abbreviated printout of "plume"	
AHNG71	A			BC	HNG7	The correction sets AHNG71 through AVALDEZ inactivate the correction set(s) indicated immediately to the left through the YANK feature to maintain the HN library oriented toward the basic Phase II run. Reversal of directives by YANK or DELETE effectively reactivates the desired features
AVARC1	A			BC	VARCORL	
AVARW1	A			BW	VARWIND	
AVARW2	A			BW	VARWINE	
ACONV1				B1	CONPRT CONPRU CONVCT CONVCU	
ACONV2	A			B1	CONPRV	
AFLAT1	A			B2	FLAT FLATA FLATIO FLATIP	
AMONT1	A			B4	MONTE	
AVALDEZ	A			BV	VALDEZ	
HNII (series)	E				Reactivates appropriate features and feature options for particular Phase II objective	

FEATURE CODE:

1	Non-linear enhancement	B	BASIC HN PHASE II
2	Layer disappearance enhancement	C	VARIABLE CORIOLIS
4	Monte Carlo extension	V	MAXDIM=1200
		W	VARIABLE WIND

TABLE 7. HN LIBRARY PHASE IV MODULES (page 1 of 2)

TYPE/ NAME	STATUS	DEPENDENCY	INCOMPATIBILITY	FEATURE	DESCRIPTION COMMENTS
COMDECK/					
A					COMMON/A/(control linkage)
B					COMMON//(advect array linkage)
C					COMMON/1/(utility linkage)
ABC					COMMON/ABC/(thermal array linkage)
THERM					COMMON/THERM/(thermal control)
DECK/					
AD	A			3	Includes routines ADVECT,BDVECT,INPOUT, IOERR,XMIT,PRTMAX
TH	A			3	Includes routines THERMAL,SETFLD,WATER ANOMALY,CALC
CORRECTION SET/					
A1	A			3	Introduced routines PRTMAX,XMIT and PRTMAX, provided 3100-6500 conversion
T1	A			3	Condensed TH routines to single routine THERMAL
COM	A			3	Restructured common decks B,ABC,THERM
A2	A	A1		3	Restructured linkage to thermal and utility routines for duplex advection limited printout
T2	A	T1		3	Adjusted computational boundaries,limited the printed output
A3	A	A2		3	Restart/data save, interpolated velocity, v direction corrections and general improvement of coding structure
T3	A	T2		3	General improvement of coding structure, comments and corrected anomaly treatment

TABLE 7 (page 2 of 2)

A4	E	A2			Replaces wind stress with true wind in m/sec units
T4	E	T3			Replaces dynamic moisture exchange computation with determination from relative humidity
HNIVTA	E			3	Dummy ident

Status Code: A-Active module on library
E-Extra-library correction set
I-Inactive module on library

Feature Code: 3-Thermal advection

TABLE 8. HN LIBRARY USAGE JOB CONTROL DECK SETUP FOR CDC6500

```
[JOBNAME],TU2.[USER NAME]
VSN(OLDPL=#1)†
REQUEST,OLDPL,MT,HI,NORING.
UPDATE,Q,L=124.        (L=124 PROVIDES ESSENTIAL PRINTED OUTPUT)
UNLOAD,OLDPL.
FTN,L=0,OPT=2,I=COMPILE.
COMMENT.  REPLACE L=0 WITH L,R=2 FOR REFERENCE LISTING.
RETURN,COMPILE.
VSN(TAPE1=#2,TAPE2=#3)
REQUEST,TAPE1,MT,HI,RING.*
REQUEST,TAPE2,MT,HI,NORING.*
LGO.
RETURN,TAPE1,TAPE2.
EXIT.
  7
  8   (binary end of record)
  9
[CDC UPDATE DIRECTIVES SET]
  7
  8   (binary end of record card)
  9
[HN PHASE CONTROL CARD SET]
  6
  7
  8   (binary end of information card)
  9
```

† Logical Units

2	Input for wind fields Phase II/output for advection fields Phase IV (as required)
1	Intermediate file 556 read/write tape
LGO	Load-and-Go unit
INPUT	System control cards, update directive sets and master control/data card decks
OUTPUT	Control card list and horizontal printed output
COMPILE	Corrected FORTRAN source code
OLDPL	Program library tape

* Tape numbers should be reversed for Phase IV jobs. (Tape2 request optional based on indicated usage.)

TABLE 9. HN PHASE I CONTROL/DATA CARD SET

```
NAME         (OPTIONAL)
GRID         (REQUIRED)
LAYS         (REQUIRED)
TIME         (OPTIONAL)
FACT         (OPTIONAL)
TIDE         (OPTIONAL, IF USED REQUIRES TIME,FACT,PRECEDING)
SORC         (OPTIONAL)
COMP         (REQUIRED)
OLDT                                   NEWT
   {symbolic Z data}| old          |{U depths}
                    | or new       |
   {U depths}       | format data  |{V depths}
                    | cards re-    |
   {V depths}       | quired       |
SETU         (OPTIONAL)
SETV         (OPTIONAL)
GRDZ         (OPTIONAL WITH OLD FORMAT-REQUIRED WITH NEW FORMAT)
SETZ         (OPTIONAL FOR OLD FORMAT-NOT REQUIRED FOR NEW FORMAT)
VOLM         (OPTIONAL)
TAPE         (REQUIRED AS THE LAST CARD OF THE SET)
```

Note: Multiple sets may be processed by Phase I. If multiple sets are written on the intermediate data set, they are separated by an end-of-file mark. User is expected to preposition intermediate data set file when stacking data sets with multiple Phase I runs.

TABLE 10. HN MASTER CONTROL CARD SET (ALL PHASES)

NAME	(OPTIONAL)
GRID	(REQUIRED)
LAYS	(REQUIRED)
TIME†	(REQUIRED)
FACT	(OPTIONAL)
TIDE	(OPTIONAL)
SORC†	(OPTIONAL)
WIND	(OPTIONAL)
FLOW	(OPTIONAL)
COMP*	(REQUIRED AS LAST CARD OF SET)

† For thermal advection applications the following control card usage variations apply:

TIME Card-The computational time step (ITINC) is replaced by the thermal advection time step which must be a multiple of the data save interval of the Phase II run which generated the intermediate data set in use.

SORC Card-The quantity of heat introduced per second is entered in place of the concentration per step; the volume per second is entered in place of volume per step. These deviations facilitate run comparisons with various steps.

WIND Card-The direction and wind speed must be floating point quantities.

* There is no required order for any control card except COMP for Phase II or thermal advection applications.

TABLE 11. HN LIBRARY USAGE BY CDC UPDATE DIRECTIVES SET

OBJECTIVE	REQUIRED UPDATE DIRECTIVE SET
Phase I (at other than BKY site)	*IDENT PHASE1 *DELETE AHNIJ1.1 *COMPILE HNI,UTIL
Phase II(Basic)†	*IDENT HNII *COMPILE HNG.UTIL
Phase IV	*IDENTI HNIVTA *COMPILE AD,TH
Phase II with Tidal Flat and Tidal Flat I/O	*IDENT HNIIFO *YANK AFLAT1 *COMPILE HNG.UTIL
Phase II with non-linear terms and non-linear I/O	*IDENT HNIICO *YANK ACONV1,ACONV2 *COMPILE HNG.UTIL
Phase II with Tide Flat non-linear and monte carlo capabilities without FLAT or CONVECTIVE I/O	*IDENT HNIIFCM *DELETE AFLAT1.1 *DELETE ACONV1.1 *DELETE AMONT1.1 (or *YANK AMONT1) *COMPILE HNG.UTIL
Phase II basic modified to 1200 array positions (for VALDEZ)	*IDENT PETROX *YANK AVALDEZ *COMPILE HNG.UTIL
Phase II basic modified to 800 array positions (for PRUDHOE)	*IDENT ICE *YANK CONTRM,BLANKA *COMPILE HNG.UTIL
Phase I (at BKY site)	*HNPHASEI *YANK AHNIJ1 *COMPILE HNI,UTIL

† The Basic HN Phase II is the three layer capability version with arrays restricted to single layer applications using ≤ 300 grid points per array.

TABLE 12. HN PHASE II TIMING PERFORMANCE CHART

JOB/ REF	BASIC HN	CONVCT	CONPRT	FLAT	FLATIO	MONTE POINTS	MODOUT	$I_{HN}\ 10^4$	REF*/COMMENT
B	X	-	-	-	-	-	240	1.18	
D	X	X	X				240	1.50	
0	X	X	X	X	X	1000	120	3.27	
1	X	X	X	X	X	1000	120	3.27	*Performance indices below the dashed line are estimates based on the references given.
2	X	X	-	X	X	1000	120	3.19	
3	X	X	-	X	-	1000	120	2.94	
4	X	X	-	X	-	1000	120	2.97	
5	X	X	-	X	-	1000	∞	2.93	
6	X	-	-	X	-	1000	∞	2.49	
7	X	-	-	X	X	1000	∞	2.65	
8	X	-	-	X	X	1000	120	2.81	
9	X	X	X	X	-	1000	120	3.03	
10	X	X	X	X	X	4000	∞	5.35	
11						1000		.95	1,10
12			X					.08	1,2(CONPRT Index=0,if MODOUT= ∞.)
13					X			.24	2,3,4 (MODOUT=120)
14							120	.03	3,4,5 Basic output
15		X						.44	5,6
16					X			.16	6,7 (MODOUT= ∞)
17							120	.16	7,8 (BASIC+ FLATIO)
18					X			.24	1,9 (MODOUT=120)
19		X						.38	2,8
20			X					.08	3,4,9
21		X	X					.32	B,D(older version CONVCT)
22	X							1.15	B,14
23				X				.37	22,11,6
24	X			X				1.54	22,23
25	X	X						1.56	22,15,19
26	X					1000		2.11	22,11

REFERENCES

1. BAUER, R. A. and A. D. Stroud. Enhanced Hydrodynamical-Numerical Model for Near-Shore Processes. In press. Environmental Protection Agency-Corvallis Environmental Research Laboratory, Corvallis, Oregon.

2. Bauer, R. A. Description of the Optimized EPRF Multi-Layer Hydrodynamical-Numerical Model, ENVPREDRSCHFAC Technical Paper No 15-74, Environmental Prediction Research Facility, Monterey, California, 1974. 66 pp.

3. Harding, J. Unpublished program notes for HNCNTR(CDC3100). Naval Environmental Prediction Research Facility, Monterey, California, 1976.

4. Bauer, Roger A. Digitization Procedure for Wave Refraction Bathymetry. Final report on Contract N00228-76-C-3121. Naval Environmental Prediction Research Facility, Monterey, California. 42 pp.

5. Bauer, R., S. Larson, T. Laevastu and A. Stroud. Linkage of the Bengtsson Limited Area Forecast Model and the Optimized Hydrodynamical-Numerical Model of the W. Hansen Type. Northwest Fisheries Center Processed Report, July 1976. Northwest Fisheries Center, National Marine Fisheries Service, Seattle, Washington.

6. Control Data Corporation. Update Reference Manual for the Control Data® Cyber 170 Series Computer Systems, Cyber 70 Series Computer Systems, 6000 Series Computer Systems and 7600 Computer Systems, Publication No 60342500 Rev G, 1975.

7. Control Data Corporation. FORTRAN Extended Version 4 Reference Manual, Publication No 60305600 Rev H, 1975.

8. Control Data Corporation. Scope 3.4 Reference Manual, Publication No

60307200 Rev L, 1975.

9. Pedersen, L. B. and L. P. Prahm. A Method for Numerical Solution of the Advection Equation $\partial C/\partial t=-\underline{V}\cdot\underline{\nabla}C$, Meteorological Institute of Denmark, as submitted to TELLUS, August, 1973. 36 pp.

APPENDIX A

PHASE I PROGRAM NOTES

DESCRIPTION OF PHASE I PROGRAM

Program LAYER3S (Blocks A-E:HNI.2,HNI.143)

This program was modified for CDC 6500 FORTRAN EXTENDED compiler from the original CDC 3100 program and retains much of the integer arithmetic logic used to reduce core requirements on the 3100. The program includes the main program LAYER3S and the routines XMIT, GRID, INPOUT, IOERR, TIDCUR, GEOPOS and PRTMAX. The main program is divided into a number of sections that process master and Phase I control cards. Although sections are activated independently, the effect varies depending on the order of the control cards. To describe the program the sections that process individual cards are grouped into the blocks shown in Figure 3. The sequence numbers refer directly to the UPDATE sequence numbers in the program listing at the end of this appendix to the right of the FORTRAN code.

This program uses the following logical units:

LU	Description
INPUT	Card input file
OUTPUT	Printed output file
PUNCH	Punched card output file
1	Intermediate data set file
2	Scratch file
3	≡ INPUT
61	≡ OUTPUT

Initialization (Block A:HNI.2,54)

This block provides the linkage to subroutines for standard control parameters in COMMON/DATA/, which also includes space saving scratch storage (IBUF) for subroutines. The major array variables are linked to other routines by blank COMMON, which does not require dimensional modification unless grids of more than 1000 positions are to be processed. Local LAYER3S arrays are specified in the DIMENSION statement and the DATA statement is used to store maximum array dimensions and constants. Additional constants are initialized in the first executable statements which are executed at the beginning of the program and after a TAPE card is processed.

All labelled control cards are read from file INPUT and printed immediately on OUTPUT in "A" format. The four character labels are matched with an index by table look up, and the index is used to branch to the appropriate section for further processing. The index key, card name and statement number for routing are included as a comment table in the program.

The remainder of the block contains a section for the decoding and reprinting of each Master control card to provide verification of the punching and scaling of each field. When present TIDE cards must be preceded by both TIME and FACT cards. After each TIDE card is interpreted by DECODE, TIDCUR is called to produce a printer plot tidal curve (see description of routine TIDCUR.) Unidentifiable cards, GRID specifications exceeding MAXDIM, or too many TIDE, WIND, SORC or FLOW cards cause early termination.

Process Bathymetric Data (Block B:HNI.55,76; HNI.85,91)

After Phase I control cards have been identified in Block A the following depth control, edit and data deck cards are processed in this logical block: NEWT, OLDT, SETU, SETV, SETZ cards and OLDT and NEWT decks. The NEWT and OLDT cards cause branching to the sections that read in the indicated type of depth card decks. After the tables are read the SETU, SETV cards are used to edit the fields in the hu and hv fields in original units. The SETZ cards can be used to edit symbolic hz if old style cards are to be punched after GRDZ processing. Usually hu and hv editing take place before a call to GRDZ which produces an hz field from the hu and hv arrays. GRDZ must be called after NEWT tables are read. No conversion or level adjustment occurs until the TAPE card is processed. Fewer than the required number of depth table cards following the OLDT or NEWT card will abort the run. Extra cards will cause an abort as an unidentified label card. Normal processing returns contol to Block A to read the next control card.

Land-Sea Boundaries (Block C: HNI.77,84)

GRDZ card is identified in Block A, routed to Block E for DECODE interpretation and then to Block C for generation of the symbolic hz field by calling GRID. On return the CFL limited time step is computed and printed for the maximum depth specified on the LAYS card and in the hu or hv field. See routine GRID for details of symbolic height field development and implied land-sea boundary verification procedures. Within GRID, GEOPOS is called to print a combined geographic display of the three fields. Normal processing returns control to Block A to read the next control card.

Auxiliary Processes (Block D:HNI.122,143)

After identification of the cards VOLM, OLDP and NEWP in Block A and interpretation in Block B, control is transferred for auxiliary processing to Block D. Both OLDP and NEWP allow subsets of the hu, hv arrays to be punched: OLDP produces an hz, hu and hv depth table deck; NEWP produces only the hu and hv decks. OLDP can only be used for single layer model fields and normally requires editing with SETZ cards to achieve the proper old style boundary setting. VOLM computes the cm^3 volume of a specified region to assist in prescribing external boundaries and transport computations. Normal processing returns control to Block A to read the next control card.

Output Processing (Block E:HNI.92,121)

Both GRDZ and TAPE cards are interpreted in this block by the DECODE function. The GRDZ card is routed to Block C for further processing. Logical unit 2 is employed in this block to avoid destroying the bathymetric data during processing of the TAPE card. If specified the routine GEOPOS is accessed through entry point POGO to print a combined hu, hv depth field in input units on the OUTPUT file. Conversion of the depth field to cm and adjustment of depth reference to mean sea level (MSL) is then accomplished and the symbolic height and hu and hv fields are printed on file OUTPUT by routine PRTMAX.

If tape output is specified all arrays are scaled by a factor of 10; negative boundary flags are removed; and routine INPOUT is called to buffer the fields out to logical unit 1 as type 1 records. The ζ, U and V initialization fields are prepared as integers scaled by 10 and routine INPOUT is again used to buffer these fields out to LU1 as time 0 type 2 records.

Normal processing results in return of control to Block A to reinitialize data set dependent constants for the next data set after writing two end of file marks and backspacing over the second on LU1. Since the bathymetric data still exist for the current data set, if no bathy data are read in a subsequent set, the same bathy data are available for reuse.

Routine TIDCUR (Block F:HNI.237,292)

The formal parameters represent actual parameters transferred from LAYER3S as follows: ITS, ITE, IOFF and MODOUT are as prescribed on the TIME card; SE provides the angular velocities from the FACT card; and PD provides the amplitude and phase information from the TIDE card. The scratch file, LU2, is used to save the major arrays and the arrays are used to store the computed tidal heights. Height computations occur for each MODOUT step starting at (IOFF-1)/MODOUT for (ITE-ITS-1)/MODOUT steps. The computed heights are printed and the variable height scales for the printer plot computed. The number of tidal constituents, input tidal parameters, tidal height scale and title are printed followed by the height scale and a line for each MODOUT step with approximate minutes scale and height indicated. A binary search technique is used to locate and print a reasonable zero time offset value for early Phase II convergence. Then the major arrays are restored and return is effected to LAYER3S.

Routine GRID (Block H:HNI.144,193)

Subroutine GRID computes the symbolic grid, hz, and checks the combined hz, hu and hv arrays for consistency in defining land and sea area boundaries. A boundary passes vertically through U points and horizontally through V points and must complete a division between groups of land and sea z grid points.

The formal parameters represent actual parameters transferred from LAYER3S as follows: LH provides access to the maximum depth values for each layer from the LAYS card adjusted to the same units as the bathymetric input under OLDT or NEWT, and ICC provides the key to call or not call routine GEOPOS for symbolic print and is used to return the maximum depth value to the main program.

The symbolic height field is set to -0 for the grid size. For each layer the following steps occur. The minimum layer depth is defined. All hu or hv points less than the minimum layer depth are set to -0. Those hu or hv equal to -0 and next to positive depth values in the same row for hu or the same column for hv are reset to -1. This last step is restricted so that it may not define a channel one cell wide. During the definitions phase of the symbolic heights, additional -0 hu or hv points are reset to -1 when channel conditions are detected. A channel exists when a pair of hu values bracketing a z point equal -0 or -1 and the corresponding z-bracketing hv points are positive, or vice versa. A -1 is set in the symbolic height field if the layer equals 1 and all of the four adjacent hu or hv values are 0 or -1. Each symbolic height field position is set to the layer number if one adjacent hu or hv value is positive. For the first layer the hu and hv fields are then saved on the scratch file, LU2. After all layers are processed the symbolic hz field shows the land-sea boundaries and the horizontal extent of each layer.

Since it is possible that erroneous land-sea boundary configurations may be generated, the boundary is tested for consistency. The only allowed conditions are that there be 0, 2 or 4 boundary points (-1) in the set of four values between four adjacent z points. A printed message occurs for each condition where 1 or 3 boundary points (-1's) are included around a z grid square. Corrections of such conditions may be made on subsequent Phase I runs using SETU and SETV cards. The testing is done for each layer prior to print of the combined symbolic z u v grid including the "land-sea" boundary for that layer. If the print is selected, routine GEOPOS is called to perform the function.

Once all layers have been processed and the symbolic height field is in final form, the hu and hv values for the first layer are read back from the scratch file. The maximum depth in the hu and hv fields is saved in ICC, and transfer is effected back to LAYER3S.

Routine GEOPOS; ENTRY POGO (Block G:HNI.194,236)

The formal parameter II provides GEOPOS the layer number for print heading from GRID. The hu, hv and symbolic hz are printed in a quasi-geographic relationship symbolically. Positive depth values at u and v points are replaced with +1; all -0 values print as 0; all -1 values in the hu or hv fields print as a vertical or horizontal line of asterisks, respectively; and -1 values in the symbolic height field print as -1. Overprinting is employed to represent properly intersecting land-sea boundary segments as lines of asterisks. Refer to the sample output for Phase I following the program listing. The overprinting on the CDC 7600 at BKY required special treatment. A library module has been included to remain compatible with the BKY site (BKYPRT) as required. GEOPOS returns control to GRID.

Entry POGO is called from LAYER3S for a single print on file OUTPUT of actual depth values for both hu and hv in a quasi-geographic structure in the same units as input under OLDT or NEWT control. A variable FORMAT (MCL) is used to handle the variable line lengths required throughout the process. Return from POGO is effected to LAYER3S.

Routine XMIT (Block I:HNG5.28,UTIL.66)

The routine XMIT is used to transfer values from a single position or array into another array. An array can be set to a constant, one array can be transferred into another or a row or column can be shifted in an array. The first parameter is the transfer from address; the second is the transfer to address. The third parameter indicates the number of values to be transferred and the fourth and fifth parameters control the respective increments of the transfer from and to addresses.

Routine PRTMAX (Block J:HNG8.127,UTIL.98)

Routine PRTMAX provides a single integer array print capability with titling. The array to be printed is specified as the first parameter. The second parameter is unused in this version but provides compatibility with Phases II and IV. The third parameter allows control over spacing of the printed array rows ("1H‸" for single spacing). The title information is specified in the fourth parameter and may include up to 80 characters of information. The fifth parameter specifies the layer of the array to be printed.

The sixth specifies the number of characters included in parameter four. The seventh parameter specifies the time associated with the array to be printed.

Arrays are printed in segments of NE row subsets of up to 15 columns. All segments except perhaps the last will print 15 column subsets. Arrays of 15 or fewer columns are printed in a single segment. Even ordinal segments will be printed on the same page as odd ordinal segments when single spacing has been selected and there are fewer than 25 rows as a paper saving feature. Array values are printed as 8 position integers. Return is effected to the main program on completion.

Routines INPOUT: IOERR (Block K:PR3.6,UTIL.63)

Routine INPOUT for Phase I provides the means of placing scaled, compressed data on the intermediate data tape file, LU1. An indicative printout occurs on entry to the routine showing the time, layer, record type, grid dimensions, model number and output record count effective with the last output operation on the intermediate data file. The format parameters specify respectively the first array to be written on the data tape, the layer, the record type requested, the time and the scale factor which has been applied prior to entry. The data are output in accordance with the data type specifications provided in Figure 2 as binary buffered output. Routine IOERR provides error processing when the buffer operation does not function normally.

LISTING FOR PHASE I SAMPLE RUN

The list is a reproduction of a PHASE I run. The first page of this listing is generated by the UPDATE processing of the library file using the list option L=124, as shown in the SCOPE control card example given in TABLE 8. The FORTRAN compiler listing of the Phase I program shows the UPDATE module identification and card numbers on the right edge. The run results begin after the PRTMAX listing and show an example run called TEST BOX MODEL. Control cards read by the program appear in the listing with the card labels at the end rather than the beginning of the line.

The reproduced listing shows the tide curve on two separate pages when in fact it is produced as continuous print lines spanning as many pages as are needed to depict the entire run time interval and the resolution of the number of seconds between saved results that are specified on the TIME card. Following the tide listing is the suggested offset for the TIDE card that would allow the model to be started with an increasing tide of near zero amplitude.

The NEWT table cards are followed by a GRDZ symbolic HTZ listing with a single land point separated from the rest of the area by lines of *** characters. The next page gives the CFL time step and volume calculation examples.

Water depths at U and V points are specified in a geographic relationship with the Z points indicated by + signs. The individual HTZ, HTU and HTV arrays are printed before the INPOUT lines printed when the type 1 and type 2 records are written to the intermediate data set.

END LAYER3S is printed when an end of file is deleted on input or an abnormal condition causes early termination.

```
IDENT    PHASE1                                          BKY T6/66 UPDATE 1.2/410-1H   09 MAR 77 01.15.25        PAGE   1

*****    *IDENT PHASE1
*****    *DELETE AHNIJ1.1
*****    *COMPILE HNI,UTIL


         MODIFICATIONS / DIRECTIVES

YANK$$$  *YANK VARWINE                                                     AVARW2        1
YANK$$$  *YANK CONPRV                                                      ACONV2        1
YANK$$$  *YANK HNG7                                                        AHNG71        1
YANK$$$  *YANK VARCORL                                                     AVARC1        1
YANK$$$  *YANK VARWIND                                                     AVARW1        1
YANK$$$  *YANK VALDEZ                                                      AVALDEZ       1
YANK$$$  *YANK CONVCT,CONVCU                                               ACONV1        1
YANK$$$  *YANK CONPRT,CONPRU                                               ACONV1        2
YANK$$$  *YANK FLAT,FLATA                                                  AFLAT1        1
YANK$$$  *YANK FLATIO,FLATIP                                               AFLAT1        2
YANK$$$  *YANK MONTE                                                       AMONT1        1
YANK$$$  *YANK BKYPRT                                                      AHNIJ1        2
```

```
      PROGRAM LAYER3S(INPUT,OUTPUT,PUNCH,TAPE1,TAPE2,TAPE3=INPUT,TAPE61=HNI.2
     *OUTPUT)                $ COMMON IZ(1000),IU(1000),IV(1000)      HNI.3
      COMMON/DATA/NE,ME,NO,MAXDIM,MAXLAY,MODEL,IBUF(1250)               HNI.4
      DIMENSION NAME(20),IC(8),ITIDE(5),TIDE(10),IWIND(8),ISORC(8),     HNJ.9
     1SORC(2)   ,NAMEZ(3),NAMEU(3),NAMEV(3) ,IN(12),TIDSPD(4) ,LH(3)    HNI.6
      DATA(NAME=4HNAME,4HGRID,4HTIME,4HFACT,4HTIDE,4HWIND,4HLAYS,4HSORC,HNI.7
     *4HFLOW,4HCOMP,4HNEWT,4HOLDT,4HGRDZ,4HTAPE,4HSETU,4HSETV,4HSETZ,4HVHNI.8
     VOLM,4HOLDP,4HNEWP),(MAXSORC=8),(MAXFLOW=10),(MAXWIND=4)           HNI.9
     2,(MAXDIM=3000),(MAXNAME=20),(MAXTIDE=4),(RAD=.01745329252)        HNI.10
     3   ,(NAMEZ=30HSYMBOLIC Z WATER HEIGHT TABLE ),(NAMEU=29HWATER DEPTHNI.11
     4H AT U POINTS (CM) ),(NAMEV=29HWATER DEPTH AT V POINTS (CM) )     HNI.12
     5,(MASK=77777777000000000000B)                                     HNJ.10
  111 ITNO=IWNO=ISNO=IFNO=0 $ MAXLAY=3                                  HNI.13
      MAXDIM=1000                                                       HNJ.11
    1 READ 1000,IC $ IF(EOF(3).NE.0.) GO TO 900                         HNJ.12
 1000 FORMAT(8A10)                                                      HNJ.13
      PRINT 1001,(IC(I),I=1,8),IC(1)                                    HNJ.14
 1001 FORMAT(4XR6,7A10,A4)                                              HNJ.15
      DO 2 I=1,MAXNAME $ IF((IC(1).AND.MASK).EQ.(NAME(I).AND.MASK))GOTO3HNJ.16
    2 CONTINUE   $ GO TO 900                                            HNI.19
    3 GO TO (1,4,5,6,10,16,18,20,21,31,51,51,180,180,160,160,160,160,160HNJ.17
     ,,160),I                                                           HNJ.18
C        KEY-CARD-BRANCH                                                HNI.21
C         1NAME1    2GRID4    3TIMES    4FACT6    5TIDE10   6WIND16     HNI.22
C         7LAYS18   8SORC20   9FLOW21  10COMP31  11NEWT51  12OLDT51     HNI.23
C        13GRDZ180 14TAPE180 15SETU160 16SETV160 17SETZ160 18VOLM160    HNI.24
C        19OLDP160 20NEWP160                                            HNI.25
C     READ COMMON CONTROL CARDS   **************************************HNI.26
    4 DECODE(80,1004,IC)NE,ME,DL,ROTANG,C,ALAT,R,AUS    $ DDL=DL*DL*.25 HNI.27
 1004 FORMAT(4X2I3,7E10.3)                                              HNI.28
      G=COS(2.*ALAT*RAD) $ G=((G*5.9E-6-.0026373)*G+1.)*980.616         HNI.29
      PRINT 1004,NE,ME,DL,ROTANG,C,ALAT,R,AUS,G                         HNI.30
      NO=NE*ME $ IF(NO.GT.MAXDIM)900,1                                  HNI.31
    5 DECODE(80,1005,IC) MODOUT,ITO,ITS,ITE,ITINC,IPO,IPMOD             HNI.32
 1005 FORMAT(4XI6,7I10)                                                 HNI.33
      PRINT 1005,MODOUT,ITO,ITS,ITE,ITINC,IPO,IPMOD $ GO TO 1           HNI.34
    6 DECODE(80,1006,IC)TIDSPD,TIDLAG,SLOPEX                            HNI.35
 1006 FORMAT(10X,7E10.3)                                                HNI.36
      PRINT 1006,TIDSPD,TIDLAG,SLOPEX $ GO TO 1                         HNI.37
   10 ITNO=ITNO+1 $ IF(ITNO.GT.MAXTIDE) GO TO 900                       HNI.38
      DECODE(80,1010,IC) ITIDE,TIDE     $ PRINT 1010,ITIDE,TIDE         HNI.39
 1010 FORMAT(4X4I3,I6,4F7.2,4F6.2,2F3.1)                                HNI.40
      CALL TIDCUR(ITS,ITE,MODOUT,TIDSPD,TIDE,ITIDE(5)) $ GO TO 1        HNI.41
   16 IWNO=IWNO+1 $ IF(IWNO.GT.MAXWIND) GO TO 900                       HNI.42
      DECODE(80,1016,IC) IWIND             $ PRINT 1016,IWIND  $ GO TO 1HNI.43
 1016 FORMAT(4X4I3,4X6I10)                                              HNI.44
   18 DECODE(80,1018,IC)ID,H1L,H2L,H3L,ALPHA,ALPH2,ALPH3,RHO1,RHO2,RHO3,HNI.45
     ,Y,Z                                                               HNI.46
 1018 FORMAT(4XI6,3F10.0,8F5.3)                                         HNI.47
      PRINT 1018,ID,H1L,H2L,H3L,ALPHA,ALPH2,ALPH3,RHO1,RHO2,RHO3,Y,Z    HNI.48
      GO TO 1                                                           HNI.49
   20 ISNO=ISNO+1 $ IF(ISNO.GT.MAXSORC) 900,22                          HNI.50
 1020 FORMAT(4X4I3,2I2,2I10,4E10.3)                                     HNI.51
   21 IFNO=IFNO+1 $ IF(IFNO.GT.MAXFLOW) GO TO 900                       HNI.52
   22 DECODE(80,1020,IC) ISORC,SORC $ PRINT 1020,ISORC,SORC $ GO TO 1   HNI.53
   31 DECODE(80,1005,IC) MODEL $ PRINT 1005,MODEL $ GO TO 1             HNI.54
C     READ PHASE I CONTROL CARDS  **************************************HNI.55
```

```
   51 DECODE(80,1051,IC)FAC,SEALEV                                       HNI.56
 1051 FORMAT(4X,F6.3,7F10.3)                                             HNI.57
C     READ IN NEW TYPE U AND V TABLES                                    HNI.58
      IF(I.EQ.12)GO TO 175  $ DO  60 N=1,NE $ MP=(ME-1)*NE+N             HNI.59
      READ 1059,(IU(I),I=N,MP,NE)                                        HNI.60
 1059 FORMAT(12I6)                                                       HNI.61
   60 PRINT 1060,N,(IU(I),I=N,MP,NE) $ PRINT 1060                        HNI.62
 1060 FORMAT(1XI3/(6X12I6))                                              HNI.63
      DO 120 N=1,NE $ MP=(ME-1)*NE+N   $ READ 1059,(IV(I),I=N,MP,NE)     HNI.64
  120 PRINT 1060,N,(IV(I),I=N,MP,NE)     $  PRINT 1060        $  GO TO 1 HNI.65
C     CORRECT TABLES                                                     HNI.66
  160 DECODE(80,1016,IC)NM,NP,MM,MP,ICC $ I=I-14                         HNI.67
      GO TO (166,166,164,400,360,300),I                                  HNI.68
  164 LM=0           $  GO TO 170                                        HNI.69
  166 LM=MAXDIM*I                                                        HNI.70
  170 NP=MAXO(NP,NM) $ MP=MAXO(MP,MM)                                    HNI.71
      IF(MM.LE.0.OR.MP.GT.ME.OR.NM.LE.0.OR.NP.GT.NE) PRINT 1170,IC       HNI.72
 1170 FORMAT(11H***ERROR***20A4)                                         HNI.73
      DO 172 M=MM,MP $ IP=LM+(M-1)*NE $ IM=IP+NM $ IP=IP+NP              HNI.74
      DO 172 I=IM,IP                                                     HNI.75
  172 IZ(I)=ICC     $  GO TO 1                                           HNI.76
C     SET Z U AND V ARRAYS BOUNDARY VALUES                               HNI.77
  173 IF(FAC.EQ.0.) FAC=1. $ LH=H1L/FAC $ LH(2)=H2L/FAC $ LH(3)=H3L/FAC HNI.78
      IF(H3L.EQ.0.) MAXLAY=2 $ IF(H2L.EQ.0.) MAXLAY=1                    HNI.79
      IF(ID.EQ.0) ID=MAXLAY $ IF(ID.NE.MAXLAY) GO TO 900                 HNI.80
      CFL=LH(ID) $ CALL GRID(LH,ICC)                                     HNI.81
      DO 174 J=1,2 $  CEL=DL/SQRT(CFL*FAC*2.*G)  $  PRINT 1174,CEL,CFL   HNI.82
  174 CFL=ICC $ GO TO 1                                                  HNI.83
 1174 FORMAT(15H MAX TIME STEP=F9.2,11H MAX DEPTH=F9.0)                  HNI.84
C    READ OLD TYPE TABLES                                                HNI.85
  175 DO 176 L=1,3 $ LM=(L-1)*MAXDIM $ LP=(ME-1)*NE+LM $ IND=13          HNI.86
      DO 176 N=1,NE $ IM=LM+N $ IP=LP+N                                  HNI.87
      DO 176 I=IM,IP,NE        $ IND=IND+1 $ IF(IND.LE.12) GO TO 176     HNI.88
      IND=1 $ READ 1059,IN     $ PRINT 1176,IN                           HNI.89
  176 IZ(I)=IN(IND)     $  GO TO 1                                       HNI.90
 1176 FORMAT(6X12I6)                                                     HNI.91
C     WRITE OUTPUT TAPE                                                  HNI.92
  180 DECODE(80,1005,IC) ICC,ICD                                         HNI.93
      IF(I.EQ.13) GO TO 173 $ REWIND 2 $ WRITE (2) IZ,IU,IV              HNI.94
      IF(ICD.GT.0) CALL POGO(1)          $  DO 181 I=1,NO                HNI.95
      IF(IU(I).GT.0)  IU(I)=AMINO(IU(I),LH(ID))*FAC+SEALEV               HNI.96
      IF(IV(I).GT.0)  IV(I)=AMINO(IV(I),LH(ID))*FAC+SEALEV               HNI.97
  181 CONTINUE $ CALL PRTMAX(IZ,4HNONE,1H ,NAMEZ,1,30,0)                 HNI.98
                 CALL PRTMAX(IU,4HNONE,1H ,NAMEU,1,29,0)                 HNI.99
                 CALL PRTMAX(IV,4HNONE,1H ,NAMEV,1,29,0)                 HNI.100
      IF(ICC.LT.0) GO TO 250                                             HNI.101
      DO 182 J=1 ,NO                                                     HNI.102
      IZ(J)=IZ(J)*10 $ IF(IZ(J).LE.0)IZ(J)=-0                            HNI.103
      IU(J)=IU(J)*10 $ IF(IU(J).LE.0)IU(J)=-0                            HNI.104
      IV(J)=IV(J)*10 $ IF(IV(J).LE.0)IV(J)=-0                            HNI.105
  182 CONTINUE $ CALL INPOUT(IZ,1,1,0,10)                                HNJ.19
C     OUTPUT INITIAL COMPUTATIONAL FIELDS FOR Z U AND V                  HNI.107
      DO 195 I=1,MAXLAY $ IF(LH(I).EQ.0) GO TO 200                       HNI.108
      L=0 $ IF(I.EQ.1) GO TO 184                                         HNI.109
      REWIND 2 $ READ(2)IZ,IU,IV $ L=LH(I-1)                             HNI.110
  184 DO 190 J=1,NO $ IF(IZ(J).LT.1) GO TO 185 $ IZ(J)=2 $ GO TO 186     HNI.111
  185 IZ(J)=-0                                                           HNI.112
```

```
PROGRAM LAYER3S  7600→7600 OPT=2                              FTN 4.6+439/040    09 MAR 77 01.15.264 BKY PAGE      3

          186 IF(IU(J).LE.L) GO TO 187 $ IU(J)=2 $ GO TO 188                        HNI.113
          187 IU(J)=-0                                                              HNI.114
          188 IF(IV(J).LE.L) GO TO 189 $ IV(J)=2 $ GO TO 190                        HNI.115
          189 IV(J)=-0                                                              HNI.116
          190 CONTINUE $ CALL INPOUT(IZ,1,2,0,10)                                   HNJ.20
          195 CONTINUE                                                              HNI.118
          200 ENDFILE1 $ ENDFILE1 $ BACKSPACE1                                      HNI.119
          250 REWIND 2 $ READ(2) IZ,IU,IV $ PRINT 1250 $ GO TO 111                  HNI.120
         1250 FORMAT(1H1)                                                           HNI.121
        C     PUNCH NEW U AND V ARRAY CARDS                                         HNI.122
          300 MM=MAXO(MM,1) $ MP=MINO(MP,ME) $ NM=MAXO(NM,1) $ NP=MINO(NP,NE)       HNI.123
              DO 310 N=NM,NP           $ MMM=(MM-1)*NE+N $ MMP=(MP-1)*NE + N        HNI.124
          310 PUNCH 1059,(IU(I),I=MMM,MMP,NE)                                       HNI.125
              DO 320 N=NM,NP           $ MMM=(MM-1)*NE+N $ MMP=(MP-1)*NE + N        HNI.126
          320 PUNCH 1059,(IV(I),I=MMM,MMP,NE)   $ GO TO 1                           HNI.127
        C     PUNCH OLD TYPE TABLES                                                 HNI.128
          360 DO 390 L=1,3 $ LM=(L-1)*MAXDIM $ IND=0                                HNI.129
              DO 370 N=NM,NP $ IM=LM+N $ DO 370 M=MM,MP                             HNI.130
              I=IM+(M-1)*NE $ IND=IND+1 $ IF(IND.LE.12) GO TO 370                   HNI.131
              PUNCH 1059,IN $ IND=1                                                 HNI.132
          370 IN(IND)=IZ(I)                                                         HNI.133
          390 PUNCH 1059,(IN(I),I=1,IND) $ GOTO 1                                   HNI.134
          400 IF(ICC.NE.0) SUM=0. $ DO 405 M=MM,MP $ MMM=(MAXO(1,M-1)-1)*NE         HNI.135
              NN=(M-1)*NE$DO405N=NM,NP $ NNM=MAXO(N-1,1) $ MMN=NN+N $ MMMN=MMM+NHNK.1
          405 SUM=SUM+FLOAT(MAXO(IU(MMN),0)+MAXO(IV(MMN),0)+MAXO(IU(MMMN),0)+       HNI.137
             *                MAXO(IV(NNM),0))   $  SUM=SUM*DDL*FAC                 HNI.138
              PRINT 1405,SUM $ GO TO 1                                              HNI.139
         1405 FORMAT( 30X16HVOLUME CUBIC CM=E11.4)                                  HNI.140
          900 PRINT 1900                                                            HNI.141
         1900 FORMAT(12H END LAYER3S)                                               HNI.142
              END                                                                   HNI.143


CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

      6      I     SYMBOLIC   HOLLERITH CONSTANT .GT. 10 CHARACTERS, EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.
      6      I     WATER DE   HOLLERITH CONSTANT .GT. 10 CHARACTERS, EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.
      6      I     WATER DE   HOLLERITH CONSTANT .GT. 10 CHARACTEPS, EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.
     80      I     LH         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
SUBROUTINE GRID      7600→7600 OPT=2                    FTN 4.6+439/040     09 MAR 77 01.15.26♦ BKY PAGE      1

      SUBROUTINE GRID(LH,ICC)                                           HNI.144
      COMMON IZ(1000),IU(1000),IV(1000)                                 HNI.145
      COMMON/DATA/NE,ME,NO,MAXDIM,MAXLAY,MODEL,IBUF(1250)               HNI.146
      DIMENSION LH(1)                                                   HNI.147
      CALL XMIT(-0,IZ,NO,0,1)                                           HNI.148
      REWIND 2                                                          HNI.149
C     SET HTU BOUNDARIES                                                HNI.150
      DO 800 II=1,MAXLAY                                                HNI.151
      LHM=0 $ IF(II.NE.1) LHM=LH(II-1)                                  HNI.152
      DO 110 N=1,NE $ MP=(ME-1)*NE+N        $ IFG=0                     HNI.153
      DO 100 I=N,MP,NE $ IF(IU(I).LE.LHM)IU(I)=-0$IF(IFG.EQ.0)GOTO80    HNI.154
      IF(IU(I).GT.0) GO TO 100 $ IU(I)=-1 $ IFG=0 $ GO TO 100           HNI.155
   80 IF(IU(I).LE.0) GO TO 100 $ IFG=-1 $ IF(I.EQ.N) GO TO 100          HNI.156
      INE=I-NE $ IU(INE)=-1                                             HNI.157
  100 CONTINUE                                                          HNI.158
  110 CONTINUE                                                          HNI.159
C     SET HTV BOUNDARIES                                                HNI.160
      DO 150 M=1,ME $ IP=(M-1)*NE $ IM=IP+1 $ IP=IP+NE $ IFG=0          HNI.161
      DO 140 I=IM,IP$ IF(IV(I).LE.LHM)IV(I)=-0 $ IF(IFG.EQ.0)GOTO130    HNI.162
      IF(IV(I).GT.0) GO TO 140 $ IV(I)=-1 $ IFG=0 $ GO TO 140           HNI.163
  130 IF(IV(I).LE.0) GO TO 140 $ IFG=-1 $ IF(I.NE.IM) IV(I-1)=-1        HNI.164
  140 CONTINUE                                                          HNI.165
  150 CONTINUE                                                          HNI.166
      DO 200 M=1,ME $ IP=(M-1)*NE                                       HNI.167
      MMM=IP-NE $ IF(MMM.LT.0) MMM=IP                                   HNI.168
C     SET HTZ BOUNDARIES                                                HNI.169
      DO 200 N=1,NE $ I=IP+N $ NM=MAXO(N-1,1)+IP    $ MM=MMM+N          HNI.170
      IF(IU(MM).EQ.0.AND.IU( I).EQ.0.AND.IV(NM).EQ.0.AND.IV( I).EQ.0)   HNI.171
     1 GO TO 200                                                        HNI.172
      IF(IU(MM).LE.0.AND.IU(I).LE.0)165,170                             HNI.173
  165 IF(IV(NM).GT.0.OR .IV(I).GT.0)GO TO 167                           HNI.174
      IF(II.EQ.1) IZ(I)=-1 $ GO TO 200                                  HNI.175
  167 IU(MM)=IU(I)=-1    $ IF(II.EQ.1.AND.I.NE.MM)IZ(MM)=-1  $  GO TO 190HNI.176
  170 IF( IV(NM).GT.0.OR.IV(I).GT.0) GO TO 190                          HNI.177
      IV(NM)=IV(I)=-1  $  IF(II.EQ.1.AND.I.NE.NM) IZ(NM)=-1             HNI.178
  190 IZ(I)=II                                                          HNI.179
  200 CONTINUE      $ IF(II.EQ.1) WRITE(2)IU,IV                         HNI.180
C     BOUNDRY ERROR TESTING                                             HNI.181
      DO 300 M=1,ME $ MP=(MINO(M+1,ME)-1)*NE $ MM=(M-1)*NE              HNI.182
      DO 300 N=1,NE $ NCNT=0 $ NN=MM+N                                  HNI.183
      IF(IU(NN).EQ.-1) NCNT=1 $ IF(IV(NN).EQ.-1)NCNT=NCNT+1             HNI.184
      NN=MM+MINO(N+1,NE) $ IF(IU(NN).EQ.-1) NCNT=NCNT+1                 HNI.185
      NN=MP+N $ IF( IV(NN).EQ.-1) NCNT=NCNT+1                           HNI.186
      IF(NCNT.EQ.1.OR.NCNT.EQ.3) PRINT 1299,N,M,NCNT                    HNI.187
 1299 FORMAT(29H ***ERROR LANDSEA ***N,M,NCNT3I4)                       HNI.188
  300 CONTINUE  $  IF(ICC.EQ.0) CALL GEOPOS(II)                         HNI.189
  800 CONTINUE                                                          HNI.190
  900 REWIND 2 $ READ (2) IU,IV $ LHM=0 $ DO 910 I=1,NO                 HNI.191
      LHM=MAXO(IU(I),IV(I),LHM)                                         HNI.192
  910 CONTINUE $ ICC=LHM $ RETURN $ END                                 HNI.193
```

```
      SUBROUTINE GEOPOS   7600→7600 OPT=2                                     FTN 4.6+439/040    09 MAR 77 01.15.26$ BKY PAGE      1

          1                SUBROUTINE GEOPOS(II)                                              HNI.194
                           COMMON IZ(1000),IU(1000),IV(1000)                                  HNI.195
                           COMMON/DATA/NE,ME,NO,MAXDIM,MAXLAY,MODEL,IBUF(15),IBU1(15),IBU2(15HNI.196
                          1),IBU3(15),IBU4(119C)                                              HNI.197
          5                DIMENSION MCL(3)                                                   HNI.198
                           DATA (MCL=29H(I5,1X1H+15(I6,1H+)//2X15I7/)),(LU=61)                HNI.199
                           PRINT 1000 $ NOP=(ME-1)/15+1 $ MF=-14                              HNJ.21
                      1000 FORMAT(1H0)                                                        HNI.201
                           DO 100 IK=1,NOP $ MF=MF+15 $ ML=MINO(MF+14,ME) $ WRITE(LU,1001) IIHNI.202
         10           1001 FORMAT(1H142X6HLAYER= I3)                                          HNI.203
                           WRITE(LU,1002)(I,I=MF,ML) $ MMF=(MF-1)*NE $ ML=(ML-1)*NE           HNI.204
                      1002 FORMAT(//3X15I6)                                                   HNI.205
                           DO 50 J=1,NE $ JK=J+MMF $ NN=ML+J $ I=0 $ DO 45 K=JK,NN,NE $ I=I+1HNI.206
                           IBU1(I)=IZ(K)+0$IBU2(I)=MINO(IU(K),1)+0$IBU3(I)=MINO(IV(K),1)+0    HNI.207
         15                IBUF(I)=1R    $  IF(IBU2(I).EQ.-1) IBUF(I)=1R*                     HNI.208
                           IBU4(I)=1H    $  IF(IBU3(I).EQ.-1) IBU3(I)=IBU4(I)=3R***           HNI.209
                        45 CONTINUE $ WRITE(LU,1045)(IBUF(K),K=1,I)                           HNI.210
                      1045 FORMAT(1H+3X15(5XR1))                                              HNI.211
                           WRITE(LU,1046)(IBUF(K),K=1,I)                                      HNI.212
         20           1046 FORMAT(4X15(5XR1))                                                 HNI.213
                           WRITE(LU,1047)J,(IBU1(K),IBU2(K),K=1,I)                            HNI.214
                      1047 FORMAT(I4,15(I3,2XI1))                                             HNI.215
                           WRITE(LU,1046)(IBUF(K),K=1,I)                                      HNI.216
                           WRITE(LU,1046)(IBUF(K),K=1,I)                                      HNI.217
         25             50 WRITE(LU,1050)(IBU4(K),IBU3(K),IBU4(K),K=1,I)                      HNI.218
                      1050 FORMAT(1H+3X15(R2,I1,R3))                                          HNI.219
                       100 CONTINUE  $ GO TO 900                                              HNI.220
                    C      ENTRY POGO PROVIDES U/V STRUCTURED OUTPUT FOR DEPTH VALUES, ...    HNI.221
                           ENTRY POGO  $  NOP=(ME+14)/15*15-14  $  PRINT 1000 $ K=15          HNI.222
         30                DO 200 MF=1,NOP,15  $  NN=MF+14  $  IF(NN.LE.ME)GO TO 140          HNI.223
                           K=ME-MF+1 $ NN=ME $ GO TO 300                                      HNJ.22
                       140 WRITE(LU,1140) II,(I,I=MF,NN) $ NN=(NN-1)*NE $ MM=(MF-1)*NE        HNI.225
                      1140 FORMAT(1H1//1X31HWATER DEPTH AT U/V POINTS LEVELI2//15I7/)         HNI.226
                           DO 150 J=1,NE $ M=MM+J $ N=NN+J                                    HNI.227
         35            150 WRITE(LU,MCL) J,(IU(I),I=M,N,NE),(IV(I),I=M,N,NE)                  HNI.228
                    C MCL  FORMAT(I5,1X1H+15(I6,1H+)//2X15I7/)  (MCL=1150)                    HNI.229
                       200 CONTINUE $ IF(K.EQ.15) GO TO 900 $ K=15                            HNI.230
                    C      STATEMENTS LINE 300 ARE EXTENDED RANGE OF DO 200                   HNI.231
                       300 ENCODE(30,1300,MCL) MCL(1),K,MCL(2),MCL(3),K,MCL(3)                HNI.232
         40           1300 FORMAT(A9,I2,R9,A3,I2,R5)                                          HNI.233
                           IF(K.NE.15) GO TO 140                                              HNI.234
                      1900 FORMAT(1HR)                                                        HNI.235
                       900 PRINT 1900 $ RETURN $ END                                          HNI.236


     CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

          6      I      (I5,1X1H  HOLLERITH CONSTANT .GT. 10 CHARACTERS. EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.
         43      I      30        NON-INNER LOOP BEGINNING AT THIS CARD IS ENTERED FROM OUTSIDE ITS RANGE.
      SUBROUTINE TIDCUR   7600-7600 OPT=2                          FTN 4.6+439/040     09 MAR 77 01.15.26s BKY PAGE      1

              SUBROUTINE TIDCUR(ITS,ITE,MODOUT,SE,PD,IOFF)    $ COMMON Z(3000)  HNI.237
              COMMON/DATA/IA(6),A(4),XS(4),JF(10)                               HNI.238
             1,IC(1232)                                                         HNJ.23
              DIMENSION  SE(1),PD(1),LINE(13)                                   HNI.240
              EQUIVALENCE (JE,MAXTIDE)                                          HNI.241
              DATA (IB=1H ),(ID=10H..........),(LINE=13(1H )),(MAXTIDE=4),      HNI.242
             *(RPD=.01745329252),(JF=97H(A1,R9)   (2A1,R8)  (A2,A1,R7)(A3,A1,R6)HNI.243
             *(A4,A1,R5)(A5,A1,R4)(A6,A1,R3)(A7,A1,R2)(A8,A1,R1)(A9,A1))        HNI.244
              REWIND 2 $ WRITE (2) Z                                            HNI.245
              T=NS=MODOUT $ IE=(ITE-ITS-1)/NS+1 $ XM=0. $ NI=(IOFF-1)/NS-1      HNI.246
              DO 25 J=1,JE $ K=J+JE  $ A(J)=PD(K)                               HNI.247
           25 CONTINUE $ TM=T/3600.                                             HNI.248
              DO 30 I=1,IE $  Z(I)=0.       $       T=FLOAT(I+NI)*TM            HNI.249
              DO 28 J=1,JE $  Z(I)=COS(((SE(J)*T-PD(J))*RPD))*A(J)+Z(I)         HNI.250
           28 CONTINUE      $  XS=ABS(Z(I))       $  IF(XS.GT.XM)  XM=XS        HNI.251
           30 CONTINUE $ TM=TM*60. $ DO  36 J=1,IE,12  $ K=J+11  $ T=J $ JJ=TM*THNI.252
           36 PRINT 1036,JJ,(Z(I),I=J,K)$S=XM/50.+.8$IF(S.GE.2.)GO TO 70$SSS=10.HNI.253
         1036 FORMAT(I12,(8X12F9.3/))                                           HNI.254
           61 SS=SSS=SSS/10.            $ LS=1                                  HNI.255
           63 IF(S.GE.SS)GO TO 69$IF(LS.EQ.3)GO TO 61$SS=SS/2.$LS=LS+1$GO TO 63 HNI.256
           69 S=SS          $ GO TO 75                                          HNI.257
           70 S=J=S                                                             HNI.258
           75 XS(4)=S*50.$ XS(3)=XS(4)*.5 $ XS(2)=-XS(3) $ XS(1)=-XS(4)         HNI.259
              PRINT 1075,JE,(SE(J),PD(J),A(J),J=1,JE),S $S=1./S                 HNI.260
         1075 FORMAT(1HQ/I12,11X6HW(D/H)5X4HK(D)4X5HH(CM)//(20X3F9.3))          HNL.1
              PRINT 1077,XS,(ID,J=1,13)                                         HNJ.24
         1077 FORMAT(65X6HHEIGHT/2(13XF9.3,3X)18X1HO3X2(16XF9.3)/6X12A10,A6)    HNL.2
        C*****NOTE THAT -IH- IS USED AS A RETURN KEY IN 200+  *******           HNJ.26
              IH=1H. $ DO 80 J=13,113,25 $ GO TO 200                            HNJ.27
           80 CONTINUE $ DO 90 I=1,IE $ J=MINO(MAX1(Z(I)*S+63.5,1.),126)        HNJ.28
              IH=1HX $ GO TO 200                                                HNI.267
           85 IF(MOD(I,3).NE.0) GO TO 88 $ T=I-1 $ JJ=T*TM                      HNI.268
              PRINT 1086,JJ,LINE $ GO TO 89                                     HNI.269
         1086 FORMAT(I6,13A10)                                                  HNI.270
           88 PRINT 1088,LINE                                                   HNI.271
         1088 FORMAT(6X13A10)                                                   HNI.272
           89 IH=IB $ IF(MOD(J+12,25).EQ.0) IH=ID $ GO TO 200                   HNJ.29
           90 CONTINUE                                                          HNJ.30
              I=J=44848 $ DO 130 K=1,18 $ XM=0. $ T=FLOAT(I)/3600.              HNI.276
              DO 100 L=1,JE  $  XM=COS((SE(L)*T-PD(L))*RPD)*A(L)+XM             HNI.277
          100 CONTINUE $ IF(J.NE.1) GO TO 101 $ PRINT 1100,XM,I $ GO TO 105     HNI.278
          101 J=J/2                                                             HNI.279
         1100 FORMAT(8H HEIGHT=E9.3,7H OFFSETI6)                                HNL.3
          105 IF(XM-.2) 110,130,120                                             HNI.281
          110 I=I+J $ GO TO 130                                                 HNI.282
          120 I=I-J                                                             HNI.283
          130 CONTINUE $ REWIND 2 $ PRINT 1130 $ READ (2) Z $ GO TO 900         HNL.4
         1130 FORMAT(1HR)                                                       HNL.5
        C     LINES 200+ ARE EXTENDED RANGE OF DO 80, 90                        HNI.285
          200 JW=(J-1)/10+1 $ JP=MOD(J-1,10)+1 $ IF(JP.EQ.10) GO TO 220         HNI.286
              IF(JP.EQ.1) GO TO 210                                             HNI.287
              ENCODE(10,JF(JP),LINE(JW)) LINE(JW),IH,LINE(JW) $ GO TO 230       HNI.288
          210 ENCODE(10,JF(JP),LINE(JW)) IH,LINE(JW) $ GO TO 230                HNI.289
          220 ENCODE(10,JF(JP),LINE(JW)) LINE(JW),IH                            HNI.290
          230 IF(IH.EQ.ID.OR.IH.EQ.1H ) GO TO 90 $ IF(IH.EQ.1HX) 85,80          HNI.291
          900 RETURN $ END                                                      HNI.292
```

```
SUBROUTINE INPOUT   7600→7600 OPT=2                    FTN 4.6+439/040    09 MAR 77 01.15.26$ BKY PAGE     1

      SUBROUTINE INPOUT(A,LAYER,ITY,IT,FACOUT)                          PR3.6
      COMMON/DATA/ NE,ME,NO,MAXDIM,MAXLAY,MODEL,IW(1250)                HNJ.1
      DIMENSION A(1)                                                    UTIL.5
      INTEGER A,FACOUT                                                  HNJ.2
      DATA(IREC=0),(M24=7777 7777B),(M12=7777B),(LEN=500)               UTIL.6
     1,(J=0)                                                            UTIL.7
C         -LAYER=READ,+LAYER=WRITE                                      UTIL.8
      PRINT 1000,IT,LAYER,ITY,NE,ME,MODEL,IREC                          UTIL.9
 1000 FORMAT(10H INPOUT       10I10)                                    UTIL.10
      ID1=SHIFT(IT,36).OR.SHIFT(IABS(LAYER),12)                         UTIL.11
      ID2=SHIFT(ITY,48).OR.SHIFT(NE,36).OR.SHIFT(ME,24).OR.MODEL        UTIL.12
      KP=MAXO(1,MINO(3,10-ITY))                                         UTILB.1
      DO 600 K=1,KP $ KK=(K-1)*MAXDIM                                   UTILA.2
C             OUTPUT                                                    UTIL.34
  400 NW=2 $ IW(1)=ID1 $ IW(2)=ID2                                      UTIL.35
      DO 500 I=1,NO $ N=A(I+KK).AND.M24                                 HNJ.3
      J=MOD(I,5)+1 $ GO TO (450,410,420,430,440),J                      UTIL.37
  410 NW=NW+1 $ IW(NW)=SHIFT(N,36) $ GO TO 470                          UTIL.38
  420 IW(NW)=IW(NW).OR.SHIFT(N,12) $ GO TO 470                          UTIL.39
  430 IW(NW)=IW(NW).OR.SHIFT(N,-12) $ NW=NW+1  $ N=N.AND.M12            UTIL.40
      IW(NW)=SHIFT(N,48) $ GO TO 470                                    UTIL.41
  440 IW(NW)=IW(NW).OR.SHIFT(N,24) $ GO TO 470                          UTIL.42
  450 IW(NW)=IW(NW).OR.N                                                UTIL.43
  470 IF(I.EQ.NO) GO TO 475    $ IF(NW.LT.LEN) GO TO 500                UTIL.44
      IF(NW.EQ.LEN.AND.J.NE.1) GO TO 500                                HNG6.42
  475 BUFFER OUT(1,1)(IW(1),IW(NW)) $ IREC=IREC+1 $ NW=2                UTIL.46
      IF(UNIT(1))500,480,490                                            UTIL.47
  480 IERR=3 $ GO TO 495                                                UTIL.48
  490 IERR=4                                                            UTIL.49
  495 CALL IOERR(IERR,IREC) $ GO TO 475                                 UTIL.50
  500 CONTINUE                                                          UTIL.51
  600 CONTINUE                                                          UTIL.52
      RETURN                                                            UTIL.53
      END                                                               UTIL.54
```

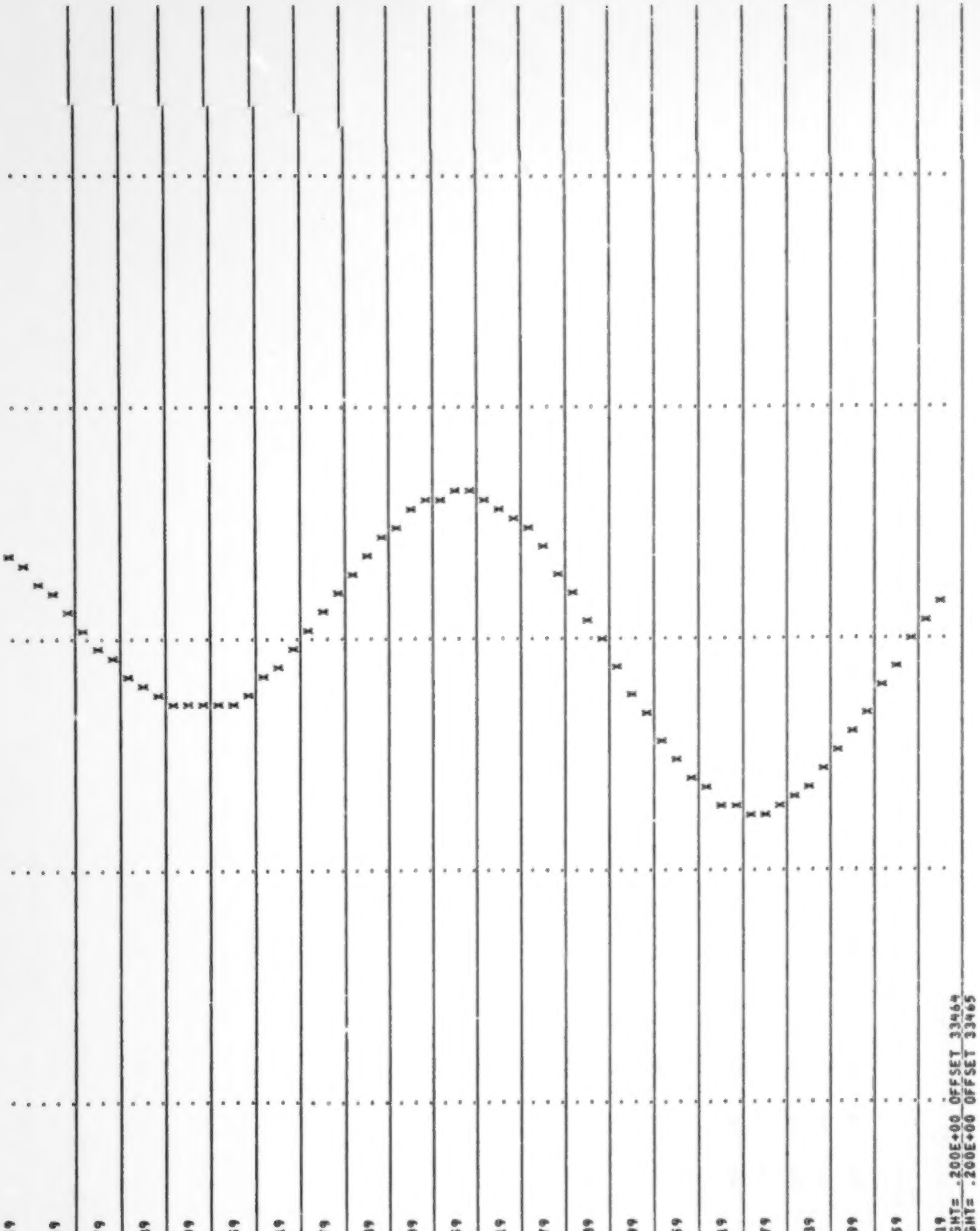
```
SUBROUTINE IOERR    7600→7600 OPT=2                    FTN 4.6+439/040    09 MAR 77 01.15.26$ BKY PAGE     1

      SUBROUTINE IOERR(L,IREC) $ DIMENSION LAB(4)                       UTIL.55
      DATA(LAB=10HEOF ERR IN,10HPAR ERR IN,10HEOT ERROUT,10HPAR ERROUT) UTIL.56
     * ,(K=0)                                                           UTILA.5
      PRINT 1000,LAB(L),IREC $ IF(L-2) 4,3,2                            UTILA.6
 1000 FORMAT(A11,I10)                                                   UTILA.7
    2 BACKSPACE 1 $ ENDFILE 1 $ BACKSPACE 1 $ IF(L.EQ.3) GO TO 4        UTILA.8
      ENDFILE 1 $ BACKSPACE 1                                           UTILA.9
    3 IREC=IREC-1 $ K=K+1 $ IF(K.LT.20) RETURN                          UTILA.10
    4 STOP                                                              UTIL.62
        END                                                             UTIL.63
```

```
SUBROUTINE XMIT      7600→7600 OPT=2                                 FTN 4.6+439/040     09 MAR 77 01.15.26$ BKY PAGE       1

      SUBROUTINE XMIT(IA,IB,N,JA,JB) $DIMENSION IA(1),IB(1)$I=J=1       HNG5.28
      DO 5 NN=1,N $ IB(J)=IA(I) $ J=J+JB $ I=I+JA                        HNG5.29
    5 CONTINUE $ RETURN                                                  HNG5.30
      END                                                                UTIL.66
```

```
SUBROUTINE PRTMAX    7600→7600 OPT=2                                 FTN 4.6+439/040     09 MAR 77 01.15.26$ BKY PAGE       1

      SUBROUTINE PRTMAX(S,D,IS,NAME1,LE,NC,IT)                           HNG8.127
C     ******** 7600 VERSION *************                               UTIL.68
      DIMENSION S(1),D(1)                                                UTIL.69
      INTEGER S                                                          HNJ.4
      COMMON/DATA/ NE,ME,NO,MAXDIM,MAXLAY,MODEL,IW(1250)                 HNJ.5
      DATA (LU=61),(MAXNAME=8),(NCPW=10)                                 HNJ.6
      NW=(NC-1)/NCPW+1                                                   UTIL.73
      NOP=(ME-1)/15+1 $ MF=-14                                           UTIL.76
      CALL XMIT(NAME1,IW,NW,1,1) $CALL XMIT(1H  ,IW(NW+1),MAXNAME-NW,0,1)HNG6.43
      DO 100 IK=1,NOP $ MF=MF+15 $ ML=MINO(MF+14,ME)                     UTIL.77
      IF(NE.LT.25.AND.MOD(IK,2).EQ.0.AND.IS.EQ.1H  ) GO TO 30            UTIL.78
   20 WRITE(LU,1000)(IW(I),I=1,MAXNAME),LE,IT                            HNG8.128
 1000 FORMAT(1H120X8A10,I3,7X5HTIME=I10)                                 HNG5.35
   30 WRITE(LU,1001)(I,I=MF,ML)                                          UTIL.81
 1001 FORMAT(//6X15I8)                                                   UTIL.82
      MMF=(MF-1)*NE $ ML=(ML-1)*NE $ IF(D(1).NE.4HNONE) GO TO 60         UTIL.83
      DO 50 J=1,NE      $ JK=J+MMF $ NN=ML+J                             UTIL.84
   50 WRITE(LU,2000)IS,J,(S(K),K=JK,NN,NE)                               UTIL.85
 2000 FORMAT(A1,I4,15I8)                                                 HNJ.7
   60 CONTINUE                                                           HNJ.8
  100 CONTINUE $ RETURN                                                  UTIL.97
      END                                                                UTIL.98
```

```
LOAD MAP.                                     LINK - BKY 6000/7000 8.4       09 MAR 77 01.15.30       PAGE    1

FL REQUIRED TO LOAD    41100
FL REQUIRED TO RUN     34300
INITIAL TRANSFER TO LAYER35   -       7726


BLOCK ASSIGNMENTS.

BLOCK         ADDRESS    LENGTH   FILE

 /DATA/           100      2350
LAYER35          2450      7325   LGO
GRID            11775       355   LGO
GEOPOS          12352       504   LGO
TIDCUR          13056       632   LGO
INPOUT          13710       230   LGO
IOERR           14140        60   LGO
XMIT            14220        20   LGO
PRTMAX          14240       257   LGO
 /STP.END/      14517         1
 /FCL.C./       14520        21
 /QLINLMT/      14541         1
 /QJOBSFL/      14542         2
 /QASAFLG/      14544         1
 /.QCOMP./      14545         1
 /Q8.IO./       14546       244
FTN4LIB         15012         0   FTN4LIB
BACKSP=         15012        62   FTN4LIB
BUFIO=          15074       144   FTN4LIB
BUFOUT=         15240        55   FTN4LIB
CIO=            15315        13   FTN4LIB
COMIO=          15330        61   FTN4LIB
DECODE=         15411       115   FTN4LIB
ENCODE=         15526       144   FTN4LIB
ENDFIL=         15672        70   FTN4LIB
EOF             15762        52   FTN4LIB
ERMSS           16034      1466   FTN4LIB
FECMSK=         17522        41   FTN4LIB
FLTIN=          17563       156   FTN4LIB
FLTOUT=         17741       315   FTN4LIB
FMTAP=          20256       372   FTN4LIB
FORSYS=         20650       761   FTN4LIB
 /QOVERL7/      21631         1
FORUTL=         21632        16   FTN4LIB
GETFIT=         21650        43   FTN4LIB
GOTOER=         21713        14   FTN4LIB
INCOM=          21727       262   FTN4LIB
 /IO.BUF./      22211       227
INPB=           22440       363   FTN4LIB
INPC=           23023       246   FTN4LIB
KODER=          23271       467   FTN4LIB
KRAKER=         23760       435   FTN4LIB
OUTB=           24415       277   FTN4LIB
OUTCOM=         24714       204   FTN4LIB
```

```
     LOAD MAP.                                            LINK - BKY 6000/7000 8.4       09 MAR 77 01.15.30        PAGE   2
     BLOCK ASSIGNMENTS.
     BLOCK         ADDRESS    LENGTH   FILE
     OUTC=           25120       235   FTN4LIB
     RDL=            25355        37   FTN4LIB
     RDO=            25414        23   FTN4LIB
     RDW=            25437       107   FTN4LIB
     REWIND=         25546        53   FTN4LIB
     SINCOS=         25621        74   FTN4LIB
     SQRT            25715        46   FTN4LIB
     SYSAID=         25763         1   FTN4LIB
     SYS=            25764        34   FTN4LIB
     SYS=1ST         26020        62   FTN4LIB
     UNIT            26102        65   FTN4LIB
     UZERO..         26167         2   FTN4LIB
     WTH=            26171        30   FTN4LIB
     WTL=            26221        27   FTN4LIB
     WTW=            26250        66   FTN4LIB
       //            26336      5670
TEST BOX MODEL                                                                       NAME
 5  533333.3333        0.     3.2E-6        42.       .003                           GRID
 5  5  .333E+05 0.          .320E-05  .420E+02  .300E-02 0.           .980E+03
 1200    7777777       0       86400       100         0        300              TIME
 1200    7777777       0       86400       100         0        300
    1      77777                         .99         1.022                           LAYS
    1     77777.       0.            0. .9900.0000.0001.0220.0000.0000.0000.000
     15.041068613.943035628.9841042     30.                                          FACT
        .150E+02  .139E+02  .290E+02  .300E+02 0.          0.
 5  5  1  5          117.     124.     359.    052.    1.0     2.0     3.0     4.0         TIDE
 5  5  1  5        0 117.00 124.00 359.00   52.00   1.00    2.00    3.00    4.000,00.0
      19           3.890     4.566    5.078    5.414    5.570    5.546    5.348    4.987    4.478    3.843    3.104    2.289

     259           1.428      .551    -.311   -1.128   -1.872   -2.514   -3.033   -3.410   -3.629   -3.681   -3.562   -3.273

     499          -2.821    -2.217   -1.479    -.629     .309    1.306    2.332    3.353    4.338    5.255    6.074    6.766

     739           7.307     7.677    7.860    7.846    7.629    7.211    6.598    5.802    4.842    3.740    2.522    1.220

     979           -.135    -1.508   -2.862   -4.164   -5.379   -6.476   -7.427   -8.207   -8.796   -9.180   -9.350   -9.304

    1219          -9.044    -8.580   -7.927   -7.103   -6.133   -5.047   -3.875   -2.651   -1.409    -.184     .989    2.079

       4           W(O/H)    K(O)     H(CM)

                  15.041  117.000    1.000
                  13.943  124.000    2.000
                  28.984  359.000    3.000
                  30.000   52.000    4.000
                    .500
                                                            HEIGHT
          -25.000                  -12.500                    0                     12.500                   25.000
.........................................................................................................................
             .                        .                        .       X                .                        .
 39          .                        .                        .        X               .                        .
             .                        .                        .         X              .                        .
             .                        .                        .          X             .                        .
             .                        .                        .          X             .                        .
 99          .                        .                        .          X             .                        .
             .                        .                        .          X             .                        .
             .                        .                        .         X              .                        .
```


159
219
279
339
399
459
519
579
639
699
759
819
879
939
999
1059
1119
1179
1239
1299
1359
1419
HEIGHT= .200E+00 OFFSET 33464
HEIGHT= .200E+00 OFFSET 33465

```
HEIGHT= .200E+00 OFFSET 33464
        99                                                                      COMP
        99
                                                                                NEWT
  1
          30    20    10     0     0
  2
          30    20    10     5     5
  3
          30    25    20    15    15
  4
          30    30    25    25    25
  5
          40    35    35    35    35

  1
          35    25    15     5     0
  2
          35    25    20    10    10
  3
          35    30    25    20    20
  4
          40    35    30    30    30
  5
          40    35    30    30    30

                                                                                GRDZ


                                           LAYER=  1

        1     2     3     4*    5
                           *
  1  1  1  1  1  1  1  1  * -1  0
                           *
     1     1     1     1  *******

  2· 1  1  1  1  1  1  1  1  1  1

     1     1     1     1     1

  3  1  1  1  1  1  1  1  1  1  1

     1     1     1     1     1

  4  1  1  1  1  1  1  1  1  1  1

     1     1     1     1     1

  5  1  1  1  1  1  1  1  1  1  1

     1     1     1     1     1
MAX TIME STEP=     2.70 MAX DEPTH=   77777.
MAX TIME STEP=   119.03 MAX DEPTH=      40.
     1  1  1  4               1
                               VOLUME CUBIC CM=  .1028E+12                      VOLM
     2  5  5  5               1
                               VOLUME CUBIC CM=  .1097E+12                      VOLM
        0         1                                                             TAPE
```

```
WATER DEPTH AT U/V POINTS LEVEL 1

     1        2        3        4        5
  1 +      30+      20+      10+      -1+       0+

     35       25       15        5       -1

  2 +      30+      20+      10+       5+       5+

     35       25       20       10       10

  3 +      30+      25+      20+      15+      15+

     35       30       25       20       20

  4 +      35+      30+      25+      25+      25+

     40       35       30       30       30

  5 +      40+      35+      35+      35+      35+

     40       35       30       30       30


                   SYMBOLIC Z WATER HEIGHT TABLE                                          1       TIME=          0

              1        2        3        4        5
  1           1        1        1        1       -1
  2           1        1        1        1        1
  3           1        1        1        1        1
  4           1        1        1        1        1
  5           1        1        1        1        1


                   WATER DEPTH AT U POINTS (CM)                                           1       TIME=          0

              1        2        3        4        5
  1        30       20       10       -1        0
  2        30       20       10        5        5
  3        30       25       20       15       15
  4        35       30       25       25       25
  5        40       35       35       35       35


                   WATER DEPTH AT V POINTS (CM)                                           1       TIME=          0

              1        2        3        4        5
  1        35       25       15        5       -1
  2        35       25       20       10       10
  3        35       30       25       20       20
  4        40       35       30       30       30
  5        40       35       30       30       30
 INPOUT             0           1           1           5           5          99           0
 INPOUT             0           1           2           5           5          99           3
 END LAYER3S
```

APPENDIX B

PHASE II PROGRAM NOTES

DESCRIPTION OF PROGRAM LAYER3S

This program was originally written for the CDC 6500 FORTRAN EXTENDED compiler (7). The program includes the main program LAYER3 and the routines SETHUV, EAR, XMIT, CXMIT, BARUV, UVCAL, UVSTAR, PRTMAX, INPOUT and IOERR. The program performs numerical computations simulating hydrodynamic processes by difference solution of the basic hydrodynamic equations described in (1).

To describe the program arbitrary logical blocks have been used which were defined in Figure 4. The sequence numbers refer directly to the UPDATE sequence numbers in the program listings at the end of this appendix to the right of the FORTRAN code.

This program uses the following logical units:

LU	Description
INPUT	Card input file
OUTPUT	Printed output file
1	Intermediate data tape file
3	≡OUTPUT

Initialization (Block A:HNG.2,78)

This block provides the linkage to subroutines for standard control parameters in COMMON/CONTROL/. To minimize the number of parameters passed in subroutine calling sequences the order of the arrays in the control common is essential as is the MAXDIM size of each. The dimension of the NNM, NNP arrays must be greater than or equal to NE. Additionally the 500 positions following

A6 on card sequence HNG8.1 are presumed destroyed through scratch usage by subroutines, so all permanent usage variables precede variable NM on the next card. COMMON// also has order dependent structure: the seven main computational arrays must be in the sequence shown and size MAXDIM $\geqslant$ NO for proper program function. A program comment shows the required dimension of the dummy array variable ZLAYK when multiple layers are desired. ZLAYK(7*MAXDIM*(K-1)) $\equiv$ Z_2(MAXDIM),...,HGV_2(MAXDIM),Z_3(MAXDIM),...,HGV_3(MAXDIM) where Z_2 represents the second layer Z array, etc. Zi, etc., will be used in subsequent computational descriptions to imply any and all layers of the indicated computation. Three single layer basic array sizes have been provided on the library with the primary (active) size being 300 positions (refer to TABLE 6). The relative position of the XK, YK arrays and MAXDIM size is also important.

After initialization of run constants and variables, each master control card is read and printed in "A" format, identified by name in the index table NAME and routed to subsections of the initialization block for DECODE interpretation and preliminary processing. The following conditions cause immediate termination of the run during master control card processing: no COMP card (end of INPUT file); unidentified control card; or too many TIDE, WIND, SORC or FLOW cards.

After normal processing of each master control card except COMP control is returned to read the next control card. Following the COMP card routine INPOUT is used to obtain the type 1 data records from LU1 for the HTZ, HTU and HTV arrays. Boundary condition 4 defined in (1) is applied to the depth fields via CXMIT call and then additional program constants are defined including the uppermost and lowermost sea positions for each column of the symbolic array for each layer. These column sea limit values are used to minimize the range of computation DO loops. The layer dependent initial or restart type 2 Z, U and V fields are read using INPOUT, constants are computed and the thicknesses HGU, HGV are computed in SETHUV. The program terminates if records matching the header computed from the INPOUT call are not found on LU1.

ζ Computation and Prescriptions (Block B:HNG8.26,133)

After initiating the time loop, routine BAR is called to compute $\overline{\zeta}$ into array ZMi; then ζi is computed into array Zi. Boundary condition 5 is implicit in the $\overline{\zeta i}$, condition 3 is applied during the computation for $\overline{U}$ and $\overline{V}$ values outside the open boundary and condition 4 is applied to Hu and Hv values outside the open boundary. Conditions 3, 4 and 5 as applied constitute the dynamic form of prescribing ζ under condition 2. Any available external tidal information replaces the dynamically computed values to give the alternate form of condition 2 prescription. Volume sources are then prescribed as modifications to positions of Zi as specified on SORC cards. The first time or any time thereafter specified for a wind to start or end the wind support section defines the wind stress effect fields in XK and YK. Any subsequent wind card must start at or subsequent to termination of the prior wind specification or at the same time as the prior wind specification. If a wind is specified to start prior to the end of a prior WIND termination specification, but after the prior wind initiation, the wind specified will not begin until the prior wind is terminated. Currently up to four constant wind subfields may be applied simultaneously, or in sequence. Modification to extend the maximum number of WIND, SORC, FLOW and TIDE cards can be accomplished by simply changing the array dimensions and the MAX___ variable associated with it in the DATA statement.

Hu, Hv Computation (Block C:HNG.134,BC8.6)

Routine SETHUV is called to perform the computation of both Hu and Hv fields into arrays HGUi and HGVi, respectively. Routine CXMIT is employed to apply boundary condition 4 at the lower boundary of Hvi fields and at the right boundary of Hui fields.

U, V Computation, Prescription (Block D:BARMOD.1,HNG8.48)

Routine UVSTAR is called to compute $\overset{*}{V}i$ into array ZSTi. $\overline{U}i$ is calculated into the Zmi array by calling routine BARUV. The routine UVCAL call computes Ui into the ZMi array, whereupon routine CXMIT is used to establish boundary condition 3 on the right boundary of the ZMi array. The ZSTi array is then used to store the $\overset{*}{U}i$ values computed by call to routine UVSTAR. The Ui values from array ZMi are transferred into array Ui by calling routine XMIT. Routine BARUV is then employed to calculate $\overline{V}i$ into the ZMi array. Vi values are computed into array Vi by calling UVCAL, whereupon boundary condition 3 on the lower boundary of Vi is effected by call to CXMIT.

Having now computed the Ui and Vi values, incremental FLOW card specifications are added to the fields. Actual velocity values could be specified through simple modification, though there has been little success in specifying velocity in this or the incremental manner due to the dearth of external information and inadequate theoretical tools.

Output Processing (Block E:HNG.191,246)

In accordance with output specification on the TIME card, routine INPOUT is called to save Zi, Ui and Vi arrays on LU1. Similarly the TIME card specifies the printed output interval for the Z and vector velocity fields. Routine PRTMAX is called for each layer to print these fields on file OUTPUT. At the end of this block determination is made whether the computation is over based on the end time on the TIME card. Normal completion of the program is indicated by END LAYER3 in the job DAYFILE. If the job ends abnormally either STOP or some other message will appear in the dayfile. If the message is not indicative as to the cause, check the input parameters and dimensions. Is NO greater than MAXDIM? Is MAXDIM equal to the dimension of all major arrays? Is NE greater than the dimension of NNP and NNM? Is ZLAYK sufficient to hold multiple layer arrays?

Routine SETHUV (Block F:HNG.249,HNG8.71)

Routine SETHUV requires no formal parameters as all required variables appear in either /CONTROL/ or //common. The routine depends on the sequence and size of HTU, HTV and HGU, HGV in the single layer case and on the entire structure of //common for multilayer cases. HTV and HGV are never referenced by name in this routine, only by index modification to name references to HTU and HGU respectively. Similarly in two layer cases Z_2, HGU_2 are not referenced except through Z, HGU by index modification. (Z_2 and HGU_2 are the first and sixth set of MAXDIM memory positions in array ZLAYK for the 2 layer case.) One call causes computation of all layers of HGU, followed by computation of all layers of HGV. No analytic boundary conditions are imposed during the thickness computation per se. The Hv computation is v-symmetric; the Hu computation is u-symmetric. Computations on the last row of Hv and the last column of Hu are replaced externally as boundary condition 5 solely for dynamic ζ computation.

Routine BAR (Block:HNG.105,BARSTAR.36)

Routine BAR requires no formal parameters and refers only to HTZ and first layer arrays Z and ZM; reference to succeeding layers when required is by index modification from the first layer array. For each position of array Zi the four nearest Zi points are used with the central value to compute a smoothed value which is returned in array ZMi. Positions over land (HTZ=0) are not defined in BAR since they are preset to zero. Further, no use of the ZMi array changes the zero values over land: if the $\overline{U}$, U or $\overline{V}$ value of the same index is defined, the ZMi point is defined, since HTZ≠0). Boundary condition 5 is effected by index substitution prior to the computation for both land-sea and open boundaries. Smoothing of the central value is interpreted as eddy diffusivity effect over the last time step. The computation is symmetric in space around the central point except at boundaries where computation is quasi-symmetric. Independent smoothing parameters may be specified at each level on the LAYS card.

Routines CXMIT, XMIT (Block H:HNG5.28,BC8.16)

Routine CXMIT provides a special controlled transfer. Routine XMIT is the same as that described in Phase I program notes. Routine CXMIT is similar to XMIT in function with the extended feature that transfer as described for XMIT occurs from A to B if and only if C is greater than D. The last parameter of CXMIT provides an index adjustment to the array C so that a control array (such as HTU) may be referenced to assure positive depth before transfer is effected. The parameters N, JA and JB function precisely as do the parameters of the same name in routine XMIT.

Routine BARUV (Block I:BARMOD.8,BARSTAR.50)

Routine BARUV is similar to routine BAR in function since the routine computes smoothed u or v velocity component fields to achieve eddy viscosity effect for the last time step. Three parameters are required: the source array, the control array and a key to indicate whether u or v type computations are being performed.

For each position of array Hi the four nearest Bi points are used with the central value to compute a smoothed value which is returned in array ZMi. Positions over land (Hi=0) are not defined in BARUV for the same reasons given in routine BAR. In actuality some values defined by the BAR routine are reset to zero by BARUV to insure lower v land-sea boundary points and right hand u land-sea boundary points remain zero. At normal land-sea boundaries, boundary condition 1 is effected. At tangential open or closed boundary points condition 6 is effected. The computation is symmetric in space except at open or closed boundaries where it is quasi-symmetric. Independent smoothing parameters may be specified at each level on the LAYS card.

Routine UVCAL (Block J:HNG8.72,94)

Routine UVCAL computes the u or v velocity component values for the next time step using two input parameters: the source array (UV) and the key (A1) to indicate u or v computations. The UVi array is used to store either the Ui or Vi computational values calculated from the ZMi, ZSTi and the HGUi (HGVi) arrays. References to HGUi also access required HGVi values using index modification that relies on the array structure of blank common. Wind stress effect terms are applied to the topmost layer of thickness greater than zero.

In the present version the friction term is constrained to be less than or equal to the smoothed velocity component. The smoothed velocity component is included in the friction term computation where a prior time step velocity component is expected. These numerical deviations showed no significant effects on experimental runs.

Boundary conditions 3 and 6 at open boundaries are implicit in the ZMi and ZSTi arrays. At closed boundaries condition 6 is applied by index modification and condition 1 is applied by default since zero velocity values persist at land-sea boundaries. Although values are computed in the last column of the Ui calculation and in the last row of the Vi computation, they have no impact on the calculation for this or future steps as these values are replaced externally prior to use in the ζ computation. The computation is quasi-symmetric in space near boundaries and symmetric otherwise, after the external boundary adjustment.

Routine UVSTAR (Block K:HNG8.95,BARSTAR.23)

The routine computes the "star" terms $\overset{*}{V}i$ and $\overset{*}{U}i$ by linearly interpolating each velocity component to the computational grid points of the normal component. The star terms are used in the coriolis and frictional effect terms of the velocity component computations. Routine UVSTAR requires three parameters designated UV, H and N. The first provides access to the variable to be interpolated, the second is no longer required and the third provides a key used to differentiate between v or u interpolation. The interpolated values from the UVi array are stored in the ZSTi array. All zero values at land-sea boundaries are used in the computation making the computation symmetric in the general case and at land-sea boundaries. The routine is quasi-symmetric at open boundaries where condition 3 is applied by index substitution.

Routine PRTMAX (Block L:HNG8.127,UTIL.98)

Routine PRTMAX prints a simple floating point array with a title row and column identification similar to PRTMAX of Appendix A. It also prints a velocity field as speed and geographic direction values separated by a comma. Seven parameters are required: S, an array of floating point values or of v velocity components; D, either an indicator that no array is present or the u velocity component; IS, the carriage control for printer vertical spacing; NAME1, the variable length title in display code; LE, the level associated with the array printed in the title line; NC, the number of display code characters NAME1; and IT, the time printed on the title line.

The array is printed in segments of 15 columns by NE rows, except the last segment which prints only to the ME column. When NE ≤ 25, ME ≥ 15 and single spacing is selected, two segments of up to 15 columns are printed on a single page.

Routines INPOUT; IOERR (Block M:PR3.6,UTIL.63)

Routine INPOUT provides access to the intermediate data set (LU1). It uses the last 500 positions of control common for an input-output buffer and depends on the blank common structure. INPOUT uses five parameters: A, the array or arrays to be processed; LAYER, the layer associated with the arrays (positive layer number indicates output; negative layer number indicates input): ITY, type of data; IT, the time associated with the arrays; and FACOUT, a factor which for most data types is used to scale the output data and to reverse the output scaling on input. Output data are packed in a compressed format with a 24 bit byte assigned to each data field, and input data are unpacked expanding the data field into a full 60 bit word for most data types. Output arrays are partitioned into 1245-byte or less sections, resulting in a maximum physical records size of 500 words. Each physical record has a two word header constructed or verified from input parameters and control common elements.

Routine IOERR provides error processing when normal access to logical unit one was not accomplished by routine INPOUT. The routine requires two parameters: L, an index which indicates the particular problem encountered; and IREC, the record number where the error occured. The routine was designed primarily to handle physical tape problems. Parity conditions should never occur when the program reads and writes to disk. The routine prints a single line to indicate the problem and the record number. Input end of file condition causes program termination after the printed line. A total of twenty input or output parity errors is allowed prior to termination. Output end of tape condition causes a backspace over the last record output, an end of file to be written, another backspace and termination.

LISTING FOR PHASE II SAMPLE RUN

The list is a reproduction of a Phase II run. The first page of the listing was produced by the UPDATE processing the library file with list option L=124 as shown in the TABLE 8 example of SCOPE control cards. The FORTRAN compiler listing of Phase II shows the UPDATE module identification and card number to the right. The run results are presented immediately following the listing of routine PRTMAX starting withthe NAME card. The NAME through COMP lines are control card images. They are followed by the INPOUT lines printed by the initialization calls to access the intermediate data set file. The following pages show the output of the time loop execution. Water heights are printed as specified on the time card with velocities following. The INPOUT line reflects the call to save data on the intermediate data set.

```
CARDS ENCOUNTERED IN INPUT                       BKY 76/66 UPDATE 1.2/410-1H   09 MAR 77 01.15.30       PAGE   1
-----------------------------------------------------------------------------------------------------------------
*****     *COMPILE HNG.UTIL

-----------------------------------------------------------------------------------------------------------------
          MODIFICATIONS / DIRECTIVES
-----------------------------------------------------------------------------------------------------------------
YANK$$$   *YANK VARWINE                                                       AVARW2        1
YANK$$$   *YANK CONPRV                                                        ACONV2        1
YANK$$$   *YANK HNG7                                                          AHNG71        1
YANK$$$   *YANK VARCOAL                                                       AVARC1        1
YANK$$$   *YANK VARWIND                                                       AVARW1        1
YANK$$$   *YANK VALDEZ                                                        AVALDEZ       1
YANK$$$   *YANK CONVCT,CONVCU                                                 ACONV1        1
YANK$$$   *YANK CONPRT,CONPRU                                                 ACONV1        2
YANK$$$   *YANK FLAT,FLATA                                                    AFLAT1        1
YANK$$$   *YANK FLATIO,FLATIP                                                 AFLAT1        2
YANK$$$   *YANK MONTE                                                         AMONT1        1
YANK$$$   *YANK HNJ                                                           AHNIJ1        1
YANK$$$   *YANK SKYPRT                                                        AHNIJ1        2
HNG       *CALL CONTRL                                                        HNG           3
HNG       *CALL BLANK                                                         HNG           7
HNG       *CALL CONTRL                                                        HNG         250
HNG       *CALL BLANK                                                         HNG         252
HNG       *CALL CONTRL                                                        HNG8         74
HNG       *CALL BLANK                                                         HNG8         76
BARSTAR   *CALL CONTRL                                                        BARSTAR       3
BARSTAR   *CALL BLANK                                                         HNG8         97
BARSTAR   *CALL CONTRL                                                        BARSTAR      25
BARSTAR   *CALL BLANK                                                         HNG8        107
BARSTAR   *CALL CONTRL                                                        BARSTAR      38
BARSTAR   *CALL BLANK                                                         HNG8        117
UTIL      *CALL CONTRL                                                        UTIL          3
UTIL      *CALL CONTRL                                                        UTIL         70
```

```
      PROGRAM LAYERS(INPUT,OUTPUT,TAPE1,TAPE3=OUTPUT)                   HNG.2
      COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,     CONTRM.1
     1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                     HNG6.1
     A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHO(3),A5RH(3),A3,A6HNG8.1
     2,NM,MM,MP,IMM,I1,I1,I2,IM,IP,TK,XW,YW,MAXNAME,MAXSORC,MAXFLOW,    HNG8.3
     3MAXTIDE,LM,LP,IC1(9),NAME(10),IW(463)                             HNG8.4
C     DIMENSION ZLAYK(MAXDIM*7*(K-1)) (WHERE K - ACTUAL NO. LAYERS).    BLANKA.1
      COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                   BLANKA.2
     A ,HGU(300),HGV(300)                                               BLANKA.3
     B ,ZLAYK(1)                                                        BLANKA.4
     W ,XK(300),YK(300)                                                 BLANKA.5
     1,COR                                                              HNG8.2
      DIMENSION IWIND(8,4),ITIDE(5,4),TIDE(10,4),IC(8)                  HNG.8
     1,TIDSPD(4),IFLOW(8,30),FLOW(4,30),ISORC(8,10),SORC(2,10),         HNGS.5
     2RHOO(1),RHO(3)                                                    HNG8.6
      EQUIVALENCE(A3,R),(A5,G),(C,C3),(IC1(2),IC),(RHOO(2),RHO)         HNG8.7
     1,(ZZ2,ZZ(2)),(ZZ3,ZZ(3))                                          HNG8.8
      DATA(MAXDIM=2544),(MAXWIND=4),(MAXTIDE=4),(MAXNAME=10),(MAXLAY=3) HNG5.9
     1,(MAXFLOW=30),(RPD=.01745329252),(ITCN=4),(MAXSORC=10),(RHOO =0.) HNG8.9
     2,(HLO=0.)                                                         HNG8.10
     2,(NAME=4HNAME,4HGRID,4HTIME,4HFACT,4HTIDE,4HWIND,4HLAYS,4HSORC,   HNG.14
     34HFLOW,4HCOMP)                                                    HNG.15
     *,(ZEROP=1)                                                        HNG81.2
C       ZEROP IS THE SMALLEST FLOATING VALUE.GT.0.                      HNG81.3
      COROLIS(ALA)=SIN(ALA*RPD)*1.4584232E-4*A1                         HNG8.11
      MINSORC=MINFLOW=99999 99999                                       HNG.18
      ITNO=ISNO=IWNO=IFNO=0                                             HNG.19
      MAXDIM=300                                                        BLANKA.6
    1 READ 1000,IC1  $ PRINT 1001,IC1 $ DO 2 I=1,MAXNAME                HNG.20
 1000 FORMAT(A4,7A10,A6)                                                HNG.21
 1001 FORMAT(1XA4,7A10,A6)                                              HNG.22
      IF(IC1(1).EQ.NAME(I)) GOTO 3                                      HNG.23
    2 CONTINUE $ GO TO 900                                              HNG.24
    3 GO TO  (1,4,5,6,10,16,18,20,21,31),I                              HNG4.3
    4 DECODE(76,1004,IC)NE,ME,DL,ROTANG,C,ALAT,R,AUS,CMPDLAT            HNG8.12
 1004 FORMAT(2I3,7E10.3)                                                HNG.27
      ROT=810.-ROTANG $ TK=ALAT*RPD $ NO=NE*ME                          HNG8.13
      TK=COS(2.*TK) $G=((TK*5.9E-6-.0026373)*TK+1.)*980.616 $ GO TO 1   HNG5.12
    5 DECODE(76,1005,IC)MODOUT,ITO,ITS,ITE,ITINC,IPO,IPMOD              HNG6.5
      IF(MODOUT.EQ.0) MODOUT=600 $ IF(IPMOD.EQ.0)IPMOD=3600 $ GO TO 1   HNG6.6
 1005 FORMAT(I6,7I10)                                                   HNG.31
    6 DECODE(76,1006,IC)TIDSPD,TIDLAG,SLOPEX                            HNG6.7
 1006 FORMAT(6X,7E10.0)                                                 HNG6.8
      DO 7 K=1,ITCN $ TIDSPD(K)=TIDSPD(K)*RPD/3600.                     HNG4.6
    7 CONTINUE      $  GO TO 1                                          HNG4.7
   10 ITNO=ITNO+1 $ IF(ITNO.GT.MAXTIDE) GO TO 900                       HNG4.8
      DECODE(76,1010,IC)(ITIDE(I,ITNO),I=1,5),(TIDE(I,ITNO),I=1,10)     HNG4.9
 1010 FORMAT(4I3,I6,4F7.2,4F6.2,2F3.1)                                  HNG4.10
      DO 11 K=1,ITCN $ TIDE(K,ITNO)=TIDE(K,ITNO)*RPD                    HNG4.11
   11 CONTINUE  $  GO TO 1                                              HNG4.12
   16 IWNO=IWNO+1 $ IF(IWNO.GT.MAXWIND) GO TO 900                       HNG4.13
      DECODE(76,1016,IC)(IWIND(I,IWNO),I=1,8) $ GO TO 1                 HNG4.14
 1016 FORMAT(4I3,4X6I10)                                                HNG4.15
   18 DECODE(76,1018,IC)HL,ALPHA,RHO $ GO TO 1                          HNG8.14
 1018 FORMAT(6X3F10.0,8F5.3)                                            HNG4.17
   20 ISNO=ISNO+1 $ IF(ISNO.GT.MAXSORC) GO TO 900                       HNG.43
      DECODE(76,1020,IC)(ISORC(I,ISNO),I=1,8),(SORC(I,ISNO),I=1,2)      HNG.44
```

```
PROGRAM LAYER2    7600-7600 OPT=2                        FTN 4.6+439/040     09 MAR 77 01.15.316 BKY PAGE       2

      MINSORC=MINO(MINSORC,ISORC(7,ISNO))$ GO TO 1                         HNG.45
   21 IFNO=IFNO+1 $ IF(IFNO.GT.MAXFLOW) GO TO 900                          HNG.46
      DECODE(76,1020,IC)(IFLOW(I,IFNO),I=1,8),(FLOW(I,IFNO),I=1,4)         HNG8.15
 1020 FORMAT  (4I3,2I2,2I10,4E10.3)                                        HNG.48
      TK=(ROT-FLOW(1,IFNO))*RPD                                            HNG6.11
      FLOW(1,IFNO)=COS(TK)*FLOW(2,IFNO)                                    HNG5.15
      FLOW(2,IFNO)=SIN(TK)*FLOW(2,IFNO)                                    HNG5.16
      MINFLOW=MINO(MINFLOW,IFLOW(7,IFNO))$ GO TO 1                         HNG.52
   31 DECODE(76,1005,IC)MODEL $ CALL INPOUT(HTZ,-1,1,0,10.)                PR[illegible]
      CALL CZMIT(HTU(NO-NE),HTU(NO),HTU(NO),0.,NE,-1,-1,-1)                [illegible]
      CALL CZMIT(HTV(NO-1 ),HTV(NO),HTV(NO),0.,ME,-NE,-NE,-NE)             8C8.2
      A1=ITINC $ A3=R*A1 $ A4=A1/DL $ A5=G*A4 $ A6=A3*.5 $C3=C*A1*10000.HNG8.16
      D1=1./(DL*DL) $ CFAC=10. $ IF(ITS.NE.0) CFAC=1000.                   PR3.2
      DO 34 M=1,ME $ IP=(M-1)*NE $ NNM(M)=IM=IP+1 $NNP(M)= IP=IP+NE        HNG.55
      DO 33 I=IM,IP                                                        HNG.56
      IF(HTZ(I).GT.0..OR.HTU(I).GT.0..OR.HTV(I).GT.0.)GO TO 32             HNG.57
      IF(NNM(M).EQ.I ) NNM(M)=I+1     $ GO TO 33                           HNG.58
   32 NNP(M)=I                                                             HNG.59
   33 CONTINUE                                                             HNG.60
   34 CONTINUE $ DO 35  L=1,MAXLAY $ IF(HL(L).LE.0.) GO TO 36              HNG8.18
      LEV(L)=JL=(L-1)*MAXDIM+7 $ CALL INPOUT(Z(JL+1),-L,2,ITS,CFAC)        PR3.3
      CALL BRIT(-0.,ZM(JL+1),MAXDIM+4,0,1)                                 HNG8.20
      A5RH(L )=RHO(L -1)/RHO(L )*A5  $ BETA(L)=.25-.25*ALPHA(L)            HNG8.21
   35 A5RHO(L )=A5-A5RH(L)  $ L=MAXLAY+1                                   HNG8.22
   36 MAXLAY=L-1                                                           HNG8.23
      COR=CORDLIS(ALAT)                                                    HNG8.24
      CFAC=1000.                                                           PR3.4
      CALL SETHUV                                                          HNG8.25
C     ********* START OF TIME LOOP *****************************************HNG.77
      ITS=ITS+ITINC     $ ITW=ITS $ MAXWIND=IWND $ IWNO=0                  HNG.78
      DO 800 IT=ITS,ITE,ITINC $ CALL BAR                                   HNG8.26
      DO 51 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 51         HNG8.27
      IMM=(M-1)*NE+1 $ DO 50 I=IM,IP $ IF(HTZ(I).LE.0.) GO TO 50 $ ZZ=0.HNG8.28
      MM=I-NE $ IF(MM.LE.0) MM=I $ NM=MAXO(I-1,IMM) $ JL=HTZ(I)            HNG8.29
   41 L=LEV(JL) $ II=L+I $ I1=L+MM $ I2=L+NM                               HNG8.30
      ZZ=(HGU(II)*U(II)-HGU(I1)*U(I1)+HGV(I2)*V(I2)-HGV(II)*V(II))+ZZ      HNG8.31
   45 Z(II)=ZM(II)-ZZ*A4 $ IF(JL.EQ.1) GO TO 50 $ JL=JL-1 $ GO TO 41       HNG8.32
   50 CONTINUE                                                             HNG.96
   51 CONTINUE                                                             HNG.97
      IF(ITNO.EQ.0) GO TO 70  $ DO 60 J=1,ITNO $ TK=FLOAT(ITIDE(5,J)+IT)HNG.98
      ZZ=0. $ DO 55 K=1,ITCN $ L=K+ITCN                                    HNG4.18
      ZZ=COS(TIDSPD(K)*TK-TIDE(K,J))*TIDE(L,J)+ZZ                          HNG4.19
   55 CONTINUE $ IF(ZZ.EQ.0.) ZZ=ZEROP $ ZZ3=ZZ*TIDE(10,J)                 HNG81.4
      ZZ2=ZZ*TIDE(9,J) $  MM=ITIDE(3,J) $ MP=ITIDE(4,J)                    HNG.104
      DO 60 M=MM,MP $ IP=(M-1)*NE $ IM=IP+ITIDE(1,J) $ IP=IP+ITIDE(2,J)  HNG.105
      DO 60 I=IM,IP $ DO 60 JL=1,MAXLAY $ L=LEV(JL)                        HNG81.5
      K=HTZ(I) $ IF(K.GE.JL.AND.ZZ(JL).NE.0.) Z(I+L)=ZZ(JL)                HNG81.6
   60 CONTINUE                                                             HNG.109
   70 IF(IT.LT.MINSORC) GO TO 90 $ DO 79 J=1,ISNO                          HNG.110
      IF(SORC(2,J).EQ.0..OR.IT.LT.ISORC(7,J).OR.IT.GT.ISORC(8,J))GOTO79  HNG.111
      MM=ISORC(3,J) $ MP=ISORC(4,J)                                        HNG.112
      DO 78 M=MM,MP $ IP=(M-1)*NE $ IM=IP+ISORC(1,J) $ IP=IP+ISORC(2,J)  HNG.113
      LM=ISORC(5,J) $ LP=ISORC(6,J) $ DO 77 I=IM,IP $ DO 77 L=LM,LP        HNG8.33
      JL=LEV(L)+I $ IF(HTZ(I).GT.FLOAT(L-1))Z(JL)=Z(JL)+SORC(2,J)*D1       HNG8.34
   77 CONTINUE                                                             HNG.118
   78 CONTINUE                                                             HNG.119
   79 CONTINUE                                                             HNG.120
   90 IF(IT.NE.ITW)  GO TO 140  $ IF(IWNO.EQ.0) GO TO 91                HNG.121
      IF(IT.EQ.IWIND(5,IWNO)) GO TO 120                                 HNG.122
   91 CALL ZMIT(0.,XK,MAXDIM*2 ,0,1) $ ITW=0                            HNG.123
  100 IWNO=IWNO+1 $ IF(IWNO.GT.MAXWIND) GO TO 140                       HNG.124
      IF(IT.GE.IWIND(6,IWNO)) GO TO 100                                 HNG6.17
      ITW = IWIND(5,IWNO)       $      IF(ITW.GT.IT) GO TO 140          HNG6.18
  120 YW=(-ROTANG-90.-FLOAT(IWIND(7,IWNO)))*RPD                         HNG81.7
      TK=IWIND(8,IWNO) $ TK=TK*TK*C3 $ XW=COS(YW )*TK $ YW=SIN(YW )*TK  HNG.127
      MM=IWIND(3,IWNO) $ MP=IWIND(4,IWNO)                               HNG.128
      DO 130 M=MM,MP $ IP=(M-1)*NE $ IM=IP+IWIND(1,IWNO)                HNG.129
      IP=IP+IWIND(2,IWNO) $ DO 130 I=IM,IP $ XK(I)=XW                   HNG.130
  130 YK(I)=YW                                                          HNG.131
      IF(IWNO.GE.MAXWIND) GO TO 135                                     HNG6.20
       IF(IT.GE.IWIND(5,IWNO+1)) GO TO 100                              HNG6.21
  135 ITW=IWIND(6,IWNO)                                                 HNG.133
  140 CALL SETHUV                                                       HNG.134
      DO 141 JL=1,MAXLAY $ K=LEV(JL)+NO                                 BC8.3
              CALL CXMIT(HGU(K-NE),HGU(K),HGU(K),0.,NE,-1,-1,-1)        BC8.4
              CALL CXMIT(HGV(K-1 ),HGV(K),HGV(K),0.,ME,-NE,-NE,-NE)     BC8.5
  141 CONTINUE                                                          BC8.6
      CALL UVSTAR(V,HTV,0) $ CALL BARUV(U,HGU,-1) $ CALL UVCAL(ZM,1.)   BARMOD.1
      DO 202 JL=1,MAXLAY $ K=LEV(JL)+NO                                 BC8.7
              CALL CXMIT(ZM(K-NE),ZM(K),HGU(K),0.,NE,-1,-1,-1)          BC8.8
  202 CONTINUE                                                          BC8.9
      CALL UVSTAR(U,HTU,1) $ DO 250 L=1,MAXLAY $ JL=LEV(L)+1            HNG8.36
  250 CALL ZMIT(ZM(JL),U(JL),NO,1,1)                                    HNG8.37
      CALL BARUV(V,HGV,0) $ CALL UVCAL(V ,-1.)                          BARMOD.2
      DO 302 JL=1,MAXLAY $ K=LEV(JL)+NO                                 BC8.10
              CALL CXMIT(V (K-1 ),V (K),HGV(K),0.,ME,-NE,-NE,-NE)       BC8.11
  302 CONTINUE                                                          BC8.12
      IF(IT.LT.MINFLOW) GO TO 500 $ DO 490 J=1,IFNO                     HNG8.39
      IF(IT.LT.IFLOW(7,J).OR.IT.GT.IFLOW(8,J)) GO TO 490                HNG8.40
      MM=IFLOW(3,J) $ MP=IFLOW(4,J) $ LM=IFLOW(5,J) $ LP=IFLOW(6,J)     HNG8.41
      TK=1. $ DO 480 L=LM,LP $ IF(L.NE.1) TK=FLOW(L+1,J)                HNG8.42
      DO 480 M=MM,MP $ IP=(M-1)*NE $ IM=IP+IFLOW(1,J) $ IP=IP+IFLOW(2,J)HNG8.43
      DO 480 I=IM,IP $ JL=LEV(L)+1                                      HNG8.44
      IF(HTU(I).GT.HLO(L)) U(JL)=U(JL)+FLOW(1,J)*TK                     HNG8.45
      IF(HTV(I).GT.HLO(L)) V(JL)=V(JL)+FLOW(2,J)*TK                     HNG8.46
  480 CONTINUE                                                          HNG8.47
  490 CONTINUE                                                          HNG8.48
  500 IF(MOD(IT,MODOUT).NE.0.OR.IT.LT.ITO) GO TO 600                    HNG.191
      DO 510 L=1,MAXLAY $ JL=LEV(L)+1                                   HNG8.49
  510 CALL INPOUT(Z(JL),L,2,IT,CFAC)                                    PR3.5
  600 CONTINUE                                                          HNG8.51
  700 IF(MOD(IT,IPMOD).NE.0.OR.IT.LT.IPO) GO TO 800                     HNG.232
      DO 705 L=1,MAXLAY $ JL=LEV(L)+1                                   HNG8.52
      CALL PRTMAX(Z(JL),4HNONE,1H ,34HWATER HEIGHT DEVIATION (CM)  LAYERHNG8.53
     1,L,34,IT)                                                         HNG8.54
  705 CALL PRTMAX(V(JL),U(JL),1H ,40HCURRENT SPEED AND DIRECTION (CM/SECHNG8.55
     1,TRUE)  LAYER ,L,40,IT)                                           HNG8.56
  800 CONTINUE                                                          HNG.246
  900 CONTINUE                                                          HNG.247
      END                                                               HNG.248
```

```
    SUBROUTINE SETHUV    7600-7600 OPT=2                            FTN 4.6+439/040    09 MAR 77 01.15.316 BKY PAGE      1

        1               SUBROUTINE SETHUV                                                   HNG.249
                        COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,       CONTRM.1
                       1ME,ND,MAIDIM,MAILAY,MODEL,NOT                                      HNG6.1
                       A,ZZ(3),HLO(3),HL(3),ALPHA(3),BETA(3),LEV(3),ASRHO(3),ASRH(3),A3,A6HNG8.1
        5              Z,IP,IM,NP,NK,KL,JL,ZZZ,IPP,IW(492)                                 HNG8.57
                 C      DIMENSION ZLAYE(MAIDIM+7+(K-1)) (WHERE K - ACTUAL NO. LAYERS).     BLANKA.1
                        COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                     BLANKA.2
                       A ,HSU(300),HSV(300)                                                 BLANKA.3
                       B ,ZLAYE(1)                                                          BLANKA.4
       10              W ,XX(300),YX(300)                                                   BLANKA.5
                       1,C10                                                                HNG8.2
                        IPP=ND & NK=NE & KL=0 & DO 120 KK=1,2                               HNG8.58
                        IF(KK.EQ.1) GO TO 40 & NK=1 & KL=MAIDIM                             HNG8.59
                     40 DO 110 M=1,NE & IM=NNM(M) & IP=NNP(M) & IF(IP.LT.IM) GO TO 110     HNG8.60
       15               IF(KK.EQ.2) IPP=M+NE                                                HNG8.61
                        DO 100 I=IM,IP & ZZ=HTU(I+KL) & IF(ZZ.LE.0.) GO TO 100              HNG8.62
                        NP=I+NK & IF(NP.GT.IPP)NP=1 & JL=MAILAY                             HNG8.63
                     60 IF(ZZ.GE.HLO(JL)) GO TO 80 & JL=JL-1 & GO TO 60                     HNG8.64
                     80 I2=LEV(JL) & I1=I2+I & I2=I2+NP & I3=I1+KL & JL=JL-1                HNG8.65
       20               ZZZ=HL(JL)-(Z(I1)+Z(I2))*.5                                         HNG8.66
                        HSU(I3)=ZZ-ZZZ & IF(JL.EQ.0) GO TO 100 & ZZ=ZZZ & GO TO 80          HNG8.67
                    100 CONTINUE                                                            HNG8.68
                    110 CONTINUE                                                            HNG8.69
                    120 CONTINUE & RETURN                                                   HNG8.70
       25               END                                                                 HNG8.71


    CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

        16      I        ZZ         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
        16      I        ZZ         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
        18      I        ZZ         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
        21      I        ZZ         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
        21      I        ZZ         ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
SUBROUTINE UVCAL    7600→7600 OPT=2                    FTN 4.6+439/040    09 MAR 77 01.15.31$ BKY PAGE     1

    1                  SUBROUTINE UVCAL(UV,A1)                                          HNG8.72
                       DIMENSION UV(1)                                                  HNG8.73
                       COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,    CONTRM.1
                      1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                    HNG6.1
    5                 A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHO(3),A5RH(3),A3,A6HNG8.1
                      2,IPP,KK,NK,JL,IK,JP,AA,I1,I2,I3,ZZ2,K,IW(488)                    HNG8.75
              C        DIMENSION ZLAYK(MAXDIM*7*(K-1)) (WHERE K - ACTUAL NO. LAYERS).   BLANKA.1
                       COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                  BLANKA.2
                      A ,HGU(300),HGV(300)                                              BLANKA.3
   10                 K ,ZLAYK(1)                                                       BLANKA.4
                      W ,XK(300),YK(300)                                                BLANKA.5
                      1,COR                                                             HNG8.2
                       IF(A1.LT.0.)GO TO 10 $ IFP=NO $ KK=0 $ NK=NE $ GO TO 20          HNG8.77
                    10 KK=MAXDIM $ NK=1                                                 HNG8.78
   15               20 DO 101 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 101   HNG8.79
                       IF(KK.NE.0) IPP=NE*M                                             HNG8.80
                       DO 100 I=IM,IP $ IK=I+KK $ IF(HTU(IK).LE.0.) GO TO 100           HNG8.81
                       DO 50 L=1,MAXLAY $ JL=MAXLAY-L+1 $ IF(HGU(LEV(JL)+IK).GT.0.)GOTO60HNG8.82
                    50 CONTINUE                                                         HNG8.83
   20               60 AA=A6 $ K=1 $ ZZ=0. $ JP=I+NK $ IF(JP.GT.IPP)JP=I                HNG8.84
                       DO 80 L=1,JL $ I2=LEV(L) $ I3=I2+IK $ I1=I2+I                    HNG8.85
                       ZZ2=A1*(Z(I2+JP)-Z(I1)) $ IF(L.EQ.JL) AA=A3                      HNG9.1
                    65 IF(HGU(I3).GT.0.) GO TO 70 $ UV(I1)=0. $ GO TO 80                HNG8.87
                    70 UV(I1)=ZM(I1)-SQRT(U(I3)*U(I3)+ZST(I1)*ZST(I1))*AA*U(I3)/HGU(I3) HNG9.2
   25                 1 -A5RH(L)*ZZ(1)-A5RHO(L)*ZZ2+ZST(I1)*A1*COR                      HNG9.3
                       IF(K.EQ.0) GO TO 80 $ UV(I1)=UV(I1)+XK(IK)/HGU(I3) $ K=0         HNG8.90
                    80 ZZ=ZZ2                                                           HNG8.91
                   100 CONTINUE                                                         HNG8.92
                   101 CONTINUE $ RETURN                                                HNG8.93
   30                  END                                                              HNG8.94


CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

    20      I       ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
    27      I       ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
SUBROUTINE UVSTAR    7600→7600 OPT=2                      FTN 4.6+439/040     09 MAR 77 01.15.31♦ BKY PAGE      1

      SUBROUTINE UVSTAR(UV,H,N) $ DIMENSION UV(1),H(1)                  HNG8.95
      COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,     CONTRM.1
     1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                     HNG6.1
     A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHO(3),A5RH(3),A3,A6HNG8.1
     2,IM,IP,IMM,IPP,I1,I2,I3,I4,JK,IW(491)                             HNG8.96
C     DIMENSION ZLAYK(MAXDIM*7*(K-1)) (WHERE K - ACTUAL NO. LAYERS).    BLANKA.1
      COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                   BLANKA.2
     A ,HGU(300),HGV(300)                                               BLANKA.3
     K ,ZLAYK(1)                                                        BLANKA.4
     W ,XK(300),YK(300)                                                 BLANKA.5
     1,COR                                                              HNG8.2
    5 DO100M=1,ME$IM=NNM(M)$IP=NNP(M)$IF(IP.LT.IM)GOTO100               BARSTAR.8
      IPP=M*NE $ IMM=IPP-NE+1                                           HNG6.32
      DO90I=IM,IP$  I1=I $ IF(N.NE.1) GO TO 10                          BARSTAR.10
      I3=MINO(I+1,IPP) $ I2=I-NE $ IF(I2.GT.0) GO TO 7                  HNG6.33
      I2=I $ I4=I3 $ GO TO 20                                           HNG6.34
    7 I4=MINO(I2+1,IMM-1) $ GO TO 20                                    HNG6.35
   10 I2=MAXO(I-1,IMM) $ I3=I+NE $ IF(I3.LE.NO) GO TO 15                HNG6.36
      I3=I $ I4=I2 $ GO TO 20                                           HNG8.98
   15 I4=MAXO(I3-1,IPP+1)                                               HNG6.38
   20 DO 80 L=1,MAXLAY $ JK=LEV(L)                                      HNG8.99
   80 ZST(JK+I)=(UV(I1+JK)+UV(I2+JK)+UV(I3+JK)+UV(I4+JK))*.25           HNG8.104
   90 CONTINUE                                                          BARSTAN.21
  100 CONTINUE $RETURN                                                  BARSTAR.22
      END                                                               BARSTAR.23
     SUBROUTINE BAR      7600→7600 OPT=2                          FTN 4.6+439/040     09 MAR 77 01.15.318 BKY PAGE       1

        1                   SUBROUTINE BAR                                                   HNG8.105
                            COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,    CONTRM.1
                           1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                    HNG6.1
                           A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHC(3),A5RH(3),A3,A6HNG8.1
        5                  2,IM,IP,MM,MP,NM,NP,IMM,IPP,JL,JK,IW(490)                         HNG8.106
                   C        DIMENSION ZLAYK(MAXDIM*7*(K-1)) (WHERE K - ACTUAL NO. LAYERS).   BLANKA.1
                            COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                  BLANKA.2
                           A ,HGU(300),HGV(300)                                              BLANKA.3
                           K ,ZLAYK(1)                                                       BLANKA.4
       10                  W ,XK(300),YK(300)                                                BLANKA.5
                           1,COR                                                             HNG8.2
                            DO 30 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 30     HNG8.108
                            IPP=M*NE $ IMM=IPP-NE+1                                          BARSTAR.29
                            DO 25 I=IM,IP $ NM=MAXO(I-1,IMM) $ NP=MINO(I+1,IPP)              HNG8.109
       15                   MM=I-NE$IF(MM.LE.0)MM=I $ MP=I+NE $ IF(MP.GT.NO)MP=I             BARSTAR.32
                            DO 20 L=1,MAXLAY $ JK=LEV(L) $ JL=JK+I                           HNG8.110
                            ZZ=L $ IF(HTZ(I).GE.ZZ) GO TO 14 $ZM(JL)=0. $ GO TO 20           BARMOD.3
                         14 IF(HTZ(NM).LT.ZZ) NM=I                                           BARMOD.4
                            IF(HTZ(NP).LT.ZZ) NP=I                                           BARMOD.5
       20                   IF(HTZ(MP).LT.ZZ) MP=I                                           BARMOD.6
                            IF(HTZ(MM).LT.ZZ) MM=I                                           BARMOD.7
                         15 ZM(JL)=Z(JL)*ALPHA(L)+(Z(NM+JK)+Z(NP+JK)+Z(MM+JK)+Z(MP+JK))*     HNG8.112
                           1BETA(L)                                                          HNG8.113
                         20 CONTINUE                                                         HNG8.114
       25                25 CONTINUE                                                         BARSTAR.34
                         30 CONTINUE $RETURN                                                 BARSTAR.35
                            END                                                              BARSTAR.36


   CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

       17      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
       17      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
       18      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
       19      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
       20      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
       21      I        ZZ        ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
SUBROUTINE BARUV    7600→7600 OPT=2                  FTN 4.6+439/040    09 MAR 77 01.15.310 BKY PAGE      1

                SUBROUTINE BARUV(B,H,N) $ DIMENSION B(1),H(1)                     BARMOD.8
                COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,     CONTRM.1
               1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                     HNG6.1
               A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),ASRHO(3),ASRH(3),A3,A6HNG8.1
               2,IM,IP,MM,MP,NM,NP,IMM,IPP,JL,JK,IW(490)                          HNG8.116
C               DIMENSION ZLAYK(MAXDIM+7*(K-1)) (WHERE K - ACTUAL NO. LAYERS).   BLANKA.1
                COMMON // Z(300),U(300),V(300),ZM(300),ZST(300)                   BLANKA.2
               A ,HGU(300),HGV(300)                                               BLANKA.3
               K ,ZLAYK(1)                                                        BLANKA.4
               W ,XK(300),YK(300)                                                 BLANKA.5
               1,COR                                                              HNG8.2
                DO 30 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 30      HNG8.118
                DO 25 I=IM,IP $ NM=MAX0(I-1,IM) $ NP=MIN0(I+1,IP)                 HNG8.119
                MM=I-NE$IF(MM.LE.0)MM=I $ MP=I+NE $ IF(MP.GT.NO)MP=I              BARSTAR.44
                DO 20 L=1,MAXLAY $ JK=LEV(L) $ JL=JK+I                            HNG8.120
                IF(H(JL).GT.0.) GO TO 15 $ ZM(JL)=0. $ GO TO 20                   HNG8.121
             15 IF(N.NE.0) GO TO 16                                               BARMOD.9
                IF(H(MM+JK).LE.0.) MM=I $ IF(H(MP+JK).LE.0.) MP=I $ GO TO 17      BARMOD.10
             16 IF(H(NM+JK).LE.0.) NM=I $ IF(H(NP+JK).LE.0.) NP=I                 BARMOD.11
             17 CONTINUE                                                          BARMOD.12
                ZM(JL)=B(JL)*ALPHA(1)+(B(NM+JK)+B(NP+JK)+B(MM+JK)+B(MP+JK))*      HNG8.124
               1BETA(1)                                                           HNG8.125
             20 CONTINUE                                                          HNG8.126
             25 CONTINUE                                                          BARSTAR.48
             30 CONTINUE $RETURN                                                  BARSTAR.49
                END                                                               BARSTAR.50
```

```
      SUBROUTINE INPOUT(A,LAYER,ITY,IT,FACOUT)                           PR3.6
      COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,      CONTRM.1
     1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                      HNG6.1
     A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHO(3),A5RH(3),A3,A6HNG8.1
     2,IW(500)                                                           UTIL.4
      DIMENSION A(1)                                                     UTIL.5
      DATA(IREC=0),(M24=7777 7777B),(M12=7777B),(LEN=500)                UTIL.6
     1,(J=0)                                                             UTIL.7
C         -LAYER=READ,+LAYER=WRITE                                       UTIL.8
      PRINT 1000,IT,LAYER,ITY,NE,ME,MODEL,IREC                           UTIL.9
 1000 FORMAT(10H INPOUT      10I10)                                      UTIL.10
      ID1=SHIFT(IT,36).OR.SHIFT(IABS(LAYER),12)                          UTIL.11
      ID2=SHIFT(ITY,48).OR.SHIFT(NE,36).OR.SHIFT(ME,24).OR.MODEL         UTIL.12
      KP=MAXO(1,MINO(3,10-ITY))                                          UTILB.1
      DO 600 K=1,KP $ KK=(K-1)*MAXDIM                                    UTILA.2
      IF(LAYER.GE.0) GO TO 400 $ FACIN=10./FACOUT*.1                     PR3.7
      NW=501 $ DO 300 I=1,NO $ IF(NW.LT.LEN) GO TO 150                   UTIL.15
      IF(NW.EQ.LEN.AND.J.NE.1) GO TO 150                                 HNG6.39
C             INPUT                                                      UTIL.17
  100 BUFFER IN(1,1)(IW(1),IW(LEN)) $ IREC=IREC+1 $ NW=2                 UTIL.18
      IF(UNIT(1))140,110,120                                             UTIL.19
  110 IERR=1 $ GO TO 130                                                 UTIL.20
  120 IERR=2                                                             UTIL.21
  130 PRINT 1130,IW(1),IW(2),ID1,ID2 $ CALL IOERR(IERR,IREC) $ GO TO 100HNG6.40
 1130 FORMAT(6(1XO20))                                                   HNG6.41
  140 CONTINUE                                                           UTILA.3
      IF(IW(1).NE.ID1.OR.IW(2).NE.ID2) GO TO 100                         UTILA.4
  150 J=MOD(I,5)+1 $ GO TO (250,210,220,230,240),J                       UTIL.24
  210 NW=NW+1 $ IWW=IW(NW) $ N=SHIFT(IWW,-36) $ GO TO 300                UTIL.25
  220 N=SHIFT(IWW,-12) $ GO TO 260                                       UTIL.26
  230 N=IWW.AND.M12 $ NW=NW+1 $ IWW=IW(NW)                               UTIL.27
      N=SHIFT(N,12).OR.SHIFT(IWW,12).AND.M12 $ GO TO 270                 UTIL.28
  240 N=SHIFT(IWW,-24)    $ GO TO 260                                    UTIL.29
  250 N=IWW                                                              UTIL.30
  260 N=N.AND.M24                                                        UTIL.31
  270 IF(N.GT.4000 0000B) N=N.OR ..NOT. M24                              UTIL.32
  300 A(KK+I)=FLOAT(N)*FACIN $ GO TO 600                                 PR3.8
C             OUTPUT                                                     UTIL.34
  400 NW=2 $ IW(1)=ID1 $ IW(2)=ID2                                       UTIL.35
      DO 500 I=1,NO $ N=A(KK+I)*FACOUT $ N=N.AND.M24                     PR3.9
      J=MOD(I,5)+1 $ GO TO (450,410,420,430,440),J                       UTIL.37
  410 NW=NW+1 $ IW(NW)=SHIFT(N,36) $ GO TO 470                           UTIL.38
  420 IW(NW)=IW(NW).OR.SHIFT(N,12) $ GO TO 470                           UTIL.39
  430 IW(NW)=IW(NW).OR.SHIFT(N,-12) $ NW=NW+1  $ N=N.AND.M12             UTIL.40
      IW(NW)=SHIFT(N,48) $ GO TO 470                                     UTIL.41
  440 IW(NW)=IW(NW).OR.SHIFT(N,24) $ GO TO 470                           UTIL.42
  450 IW(NW)=IW(NW).OR.N                                                 UTIL.43
  470 IF(I.EQ.NO) GO TO 475     $ IF(NW.LT.LEN) GO TO 500                UTIL.44
      IF(NW.EQ.LEN.AND.J.NE.1) GO TO 500                                 HNG6.42
  475 BUFFER OUT(1,1)(IW(1),IW(NW)) $ IREC=IREC+1 $ NW=2                 UTIL.46
      IF(UNIT(1))500,480,490                                             UTIL.47
  480 IERR=3 $ GO TO 495                                                 UTIL.48
  490 IERR=4                                                             UTIL.49
  495 CALL IOERR(IERR,IREC) $ GO TO 475                                  UTIL.50
  500 CONTINUE                                                           UTIL.51
  600 CONTINUE                                                           UTIL.52
      RETURN                                                             UTIL.53
      END
```

```
SUBROUTINE IOERR    7600→7600 OPT=2                        FTN 4.6+439/040    09 MAR 77 01.15.31$ BKY PAGE      1

               SUBROUTINE IOERR(L,IREC) $ DIMENSION LAB(4)                        UTIL.55
               DATA(LAB=10HEOF ERR IN,10HPAR ERR IN,10HEOT ERROUT,10HPAR ERROUT)  UTIL.56
              * ,(K=0)                                                            UTILA.5
               PRINT 1000,LAB(L),IREC $ IF(L-2) 4,3,2                             UTILA.6
          1000 FORMAT(A11,I10)                                                    UTILA.7
             2 BACKSPACE 1 $ ENDFILE 1 $ BACKSPACE 1 $ IF(L.EQ.3) GO TO 4         UTILA.8
               ENDFILE 1 $ BACKSPACE 1                                            UTILA.9
             3 IREC=IREC-1 $ K=K+1 $ IF(K.LT.20) RETURN                           UTILA.10
             4 STOP                                                               UTIL.62
                 END                                                              UTIL.63
```

```
SUBROUTINE XMIT     7600→7600 OPT=2                        FTN 4.6+439/040    09 MAR 77 01.15.31$ BKY PAGE      1

               SUBROUTINE XMIT(IA,IB,N,JA,JB) $DIMENSION IA(1),IB(1)$I=J=1        HNG5.28
               DO 5 NN=1,N $ IB(J)=IA(I) $ J=J+JB $ I=I+JA                        HNG5.29
             5 CONTINUE $ RETURN                                                  HNG5.30
               END                                                                UTIL.66
```

```
SUBROUTINE CXMIT    7600→7600 OPT=2                        FTN 4.6+439/040    09 MAR 77 01.15.31$ BKY PAGE      1

               SUBROUTINE CXMIT(A,B,C,D,N,JA,JB,JC) $ DIMENSION A(1),B(1),C(1)    BC8.13
               I=J=K=1 $ DO 5 NN=1,N $ IF(C(K).GT.D) B(J)=A(I)                    BC8.14
               K=K+JC $ J=J+JB $ I=I+JA                                           BC8.15
             5 CONTINUE $ RETURN $ END                                            BC8.16
```

```
SUBROUTINE PRTMAX    7600-7600 OPT=2                         FTN 4.6+439/040    09 MAR 77 01.15.316 BKY PAGE       1

    1               SUBROUTINE PRTMAX(S,D,IS,NAME1,LE,NC,IT)                               HNG8.127
              C     ******** 7600 VERSION *************                                    UTIL.68
                    DIMENSION S(1),D(1)                                                    UTIL.69
                    COMMON/CONTROL/HTZ(300),HTU(300),HTV(300),NNP(20),NNM(20),NE,          CONTRM.1
    5              1ME,NO,MAXDIM,MAXLAY,MODEL,ROT                                          HNG6.1
                   A,ZZ(3),HLO(1),HL(3),ALPHA(3),BETA(3),LEV(3),A5RHO(3),A5RH(3),A3,A6HNG8.1
                   1,IK,IWW,JK,I,A,B,NOP,MF,ML,MMF,NN,NW,IW(488)                           UTIL.71
                    DATA(LU=3) ,(MAXNAME= 8) ,(NCPW=10)                                    UTIL.72
                    NW=(NC-1)/NCPW+1                                                       UTIL.73
   10               NOP=(ME-1)/15+1 $ MF=-14                                               UTIL.76
                    CALL XMIT(NAME1,IW,NW,1,1) $CALL XMIT(1H ,IW(NW+1),MAXNAME-NW,0,1)HNG6.43
                    DO 100 IK=1,NOP $ MF=MF+15 $ ML=MINO(MF+14,ME)                         UTIL.77
                    IF(NE.LT.25.AND.MOD(IK,2).EQ.0.AND.IS.EQ.1H ) GO TO 30                 UTIL.78
                 20 WRITE(LU,1000)(IW(I),I=1,MAXNAME),LE,IT                                HNG8.128
   15          1000 FORMAT(1H120X8A10,I3,7X5HTIME=I10)                                     HNG5.35
                 30 WRITE(LU,1001)(I,I=MF,ML)                                              UTIL.81
               1001 FORMAT(//6X15I8)                                                       UTIL.82
                    MMF=(MF-1)*NE $ ML=(ML-1)*NE $ IF(D(1).NE.4HNGNE) GO TO 60             UTIL.83
                    DO 50 J=1,NE        $ JK=J+MMF $ NN=ML+J                               UTIL.84
   20            50 WRITE(LU,2000)IS,J,(S(K),K=JK,NN,NE)                                   UTIL.85
               2000 FORMAT(A1,I4,3X15F8.0)                                                 UTIL.86
                    GO TO 100                                                              UTIL.87
                 60 DO 90 J=1,NE $ NN=10 $ DO 80 K=MMF,ML,NE $ JK=J+K $ NN=NN+1            HNG6.44
                    MM=JK-NE $ IF(MM.LE.0) MM=JK $ NM=JK-1 $ IF(J.EQ.1) NM=JK              HNG81.8
   25               X=(D(JK)+D(MM))*.5 $ Y=(S(JK)+S(NM))*.5                                HNG81.9
                    IF(X.NE.0..OR.Y.NE.0.)GO TO 70 $ IW(NN)=1H  $ GO TO 80                 HNG81.10
                 70 A=SQRT(X*X+Y*Y) $ B=AMOD(ROT-ATAN2(Y,X)*57.295779513,360.)             HNG81.11
                    ENCODE(9,3000,IWW)A,B $ IW(NN)=IWW.OR.770000000000B                    UTIL.92
               3000 FORMAT(F5.0,F4.0)                                                      UTIL.93
   30            80 CONTINUE                                                               UTIL.94
                 90 WRITE(LU,4000)IS,J,(IW(K),K=11,NN)                                     HNG6.46
               4000 FORMAT(A1,I4,3X15A8)                                                   UTIL.96
                100 CONTINUE $ RETURN                                                      UTIL.97
                    END                                                                    UTIL.98


CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

    12     I                   CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
    16     I                   CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
```

```
LOAD MAP.                                                      LINK - BKY 6000/7000 8.4       09 MAR 77 01.15.35       PAGE    1
 FL REQUIRED TO LOAD    36500
 FL REQUIRED TO RUN     31100
 INITIAL TRANSFER TO LAYER3      -      6150


 BLOCK ASSIGNMENTS.

 BLOCK             ADDRESS    LENGTH   FILE

  /CONTROL/            100      2677
 LAYER3               2777      6066   LGO
 SETHUV              11065       102   LGO
 UVCAL               11167       161   LGO
 UVSTAR              11350       103   LGO
 BAR                 11453       106   LGO
 BARUV               11561       112   LGO
 INPOUT              11673       401   LGO
 IOERR               12274        60   LGO
 XMIT                12354        20   LGO
 CXMIT               12374        27   LGO
 PRTMAX              12423       377   LGO
  /STP.END/          13022         1
  /FCL.C./           13023        21
  /QLINLMT/          13044         1
  /QJOBSFL/          13045         2
  /QASAFLG/          13047         1
  /.QCOMP./          13050         1
  /Q8.IO./           13051       244
 FTN4LIB             13315         0   FTN4LIB
 ATAN2               13315       100   FTN4LIB
 BACKSP=             13415        62   FTN4LIB
 BUFIN=              13477        61   FTN4LIB
 BUFIO=              13560       144   FTN4LIB
 BUFOUT=             13724        55   FTN4LIB
 CIO=                14001        13   FTN4LIB
 COMIO=              14014        61   FTN4LIB
 DECODE=             14075       115   FTN4LIB
 ENCODE=             14212       144   FTN4LIB
 ENDFIL=             14356        70   FTN4LIB
 ERMS$               14446      1466   FTN4LIB
 FECMSK=             16134        41   FTN4LIB
 FLTIN=              16175       156   FTN4LIB
 FLTOUT=             16353       315   FTN4LIB
 FMTAP=              16670       372   FTN4LIB
 FORSYS=             17262       761   FTN4LIB
  /QOVERL7/          20243         1
 FORUTL=             20244        16   FTN4LIB
 GETFIT=             20262        43   FTN4LIB
 GOTOER=             20325        14   FTN4LIB
 INCOM=              20341       262   FTN4LIB
 INPC=               20623       246   FTN4LIB
 KODER=              21071       467   FTN4LIB
 KRAKER=             21560       435   FTN4LIB
```

```
         LOAD MAP.                                              LINK - BKY 6000/7000 8.4           09 MAR 77 01.15.35         PAGE     2
         BLOCK ASSIGNMENTS.

         BLOCK            ADDRESS     LENGTH    FILE

         OUTCOM=          22215        204    FTN4LIB
         OUTC=            22421        235    FTN4LIB
         RDL=             22656         37    FTN4LIB
         RDO=             22715         23    FTN4LIB
         RDW=             22740        107    FTN4LIB
         SINCOS=          23047         74    FTN4LIB
         SQRT             23143         46    FTN4LIB
         SYSAID=          23211          1    FTN4LIB
         SYS=             23212         34    FTN4LIB
         SYS=AID          23246          7    FTN4LIB
         SYS=1ST          23255         62    FTN4LIB
         UNIT             23337         65    FTN4LIB
         UZERO..          23424          2    FTN4LIB
         WTH=             23426         30    FTN4LIB
         WTL=             23456         27    FTN4LIB
         WTW=             23505         66    FTN4LIB
          //              23573       5216
NAME TEST BOX MODEL
GRID  5  533333.3333        0.    3.2E-6       42.      .003
LAYS     1     77777                     .99          1.022
TIME   600       600         0      2400       100         0      1200
FACT      15.041068613.943035628.9841042   30.
TIDE  1  1  1  4 33465   117.   124.  359.   052.   1.0   2.0   3.0   4.0
TIDE  5  5  1  1 32565   117.   124.  359.   052.   1.0   2.0   3.0   4.0
TIDE  2  5  5  5 31665   117.   124.  359.   052.   1.0   2.0   3.0   4.0
SORC  2  2  2  2 1 1       100  1000000      100.                  2.        2.
COMP    99
INPOUT            0        -1         1         5         5        99          0
INPOUT            0        -1         2         5         5        99          3
INPOUT          600         1         2         5         5        99          6
INPOUT         1200         1         2         5         5        99          9
```

```
                WATER HEIGHT DEVIATION (CM)  LAYER                                              1      TIME=     1200

              1      2      3      4      5
     1        1.     1.     1.     1.     0.
     2        1.     1.     0.    -0.    -0.
     3        1.     1.     0.    -1.    -0.
     4        1.     1.     0.    -0.    -0.
     5        0.     1.     0.     0.    -0.

                CURRENT SPEED AND DIRECTION (CM/SEC,TRUE)  LAYER                                1      TIME=     1200

              1        2        3        4        5
     1     7;182   10;181   13;181    9;181
     2     8;142   10;150   11;158    8;166    2; 96
     3     9;125    9;130    8;132    4;147    1;245
     4     9;116    8;114    7;118    2;117    0;226
     5     8;111    8;109    6;114    2;106    0;124
INPOUT         1800            1            2            5          5          99          12
INPOUT         2400            1            2            5          5          99          15



                WATER HEIGHT DEVIATION (CM)  LAYER                                              1      TIME=     2400

              1      2      3      4      5
     1        2.     2.     2.     2.     0.
     2        2.     2.     1.     1.     1.
     3        2.     1.     1.     0.     1.
     4        2.     1.     1.     1.     1.
     5        1.     1.     1.     1.     1.

                CURRENT SPEED AND DIRECTION (CM/SEC,TRUE)  LAYER                                1      TIME=     2400

              1        2        3        4        5
     1    10;184   11;183   11;182    8;181
     2    10;160   11;164   10;167    8;157    5; 92
     3    10;138   10;142   10;145    8;142    4; 97
     4    10;125   10;126    9;130    6;128    3;101
     5    10;120   10;120    8;125    5;120    3;101
```

APPENDIX C

NON-LINEAR MODIFICATIONS

CONVCT AND CONVCU SERIES

The CONVCT and CONVCU modifications add non-linear terms to Phase II u and v velocity computations. The procedure expands the logic in the BARUV routine to calculate the non-linear terms when the routine is entered through the special entry CONUV, which sets the flag ICON. CONUV is called once for each component, and the results are added to the previously computed velocity component in array ZMi. To save the required work arrays in the computational block in the main program, the UVCAL call is changed so the Vi values are computed into the ZMi array rather than into the Vi array, and an XMIT is added to transfer ZMi into Vi at the end of the calculation. Control common is expanded with the insertion of the constant A7 and variable CY before the 500 scratch positions. The computation of A7 has been added to the initialization block. CY is set in CONUV to the maximum absolute value of the non-linear component and printed in the output section as a debugging aid that has been retained as a useful feature in the model.

Boundary conditions have been presented in (1). Boundary condition 1 is applied by the default value at the land-sea boundary normal to flow and interesected by tangential flow line. Boundary condition 3 is applied at open normal boundaries by index substitution. Boundary condition 7 is applied prior to the computation by scaling the terms for which the BARUV default boundary condition 6 has applied index substitution at tangential open or closed boundaries or normal boundaries not intersected by the tangential flow line. An indexing technique is used to accomplish the computation for both the u and v non-linear term cases. The computation is symmetric or quasi-symmetric in space depending on the influence of boundary conditions.

The names of these correction sets are in alphabetical sequence. Further correction sets related to these computations would be named CONVCV, etc. This naming technique will occur automatically if all related correction sets were preceded by *IDENT CONVCT due to renaming conventions of UPDATE(6) in a duplicate name situation.

CONPRT, CONPRU AND CONPRV SERIES

The CONPRT, CONPRU and CONPRV modifications provide an output capability for the non-linear term computations. There are two separate procedures: printed output with other scheduled printed output is provided when CONPRT is used alone, and output to the intermediate data set as type 10 data records without printed output when all three modifications are used. Use of the CONPRT feature alone modifies the format for print of all single arrays so that normal printed display of layer surface deviations is in exponential format. An additional array of CZ (300) is inserted immediately following the CY variable in control common. CZi must be modified to be at least MAXDIM* (number of layers) when MAXDIM is changed and these modifications are used. The initialization block is affected by the change in control common and by a call to XMIT to initialize array CZi to zero. The u and v computational block is modified by a call to INPOUT to save the array CZi on the intermediate data set (or call to PRTMAX for CONPRT alone to print the array CZi). The output block is similarly modified by a call to INPOUT (or PRTMAX). Routine BARUV is expanded to define the CZi arrays as the computation of the non-linear terms progresses. Naming conventions are similar to those employed for CONVCT series corrections. These modifications may never be used without the CONVCT series.

LISTING FOR NON-LINEAR CORRECTION SETS

The following listing of the CONVCT and CONPRT series correction sets is in card image form as originally entered into the library. The spacing between correction sets is artificial and the sets are presented in a logical order rather than the order in which they were entered on the library. These correction sets are applied to the basic Phase II program shown in Figure 8. Since the correction sets are actually a part of the library the user is not normally concerned with these decks. They are presented in this report to document the complete model since it would be prohibitive to show all possible FORTRAN compiler listings for the different combinations of selectable options.

```
*IDENT CONVCT
*INSERT HNG8.1
     B ,A7
*INSERT HNG8.17
      A7=A4*.5
*INSERT BARMOD.1
      CALL CONUV(U,HGU,-1)
*DELETE BARMOD.2
      CALL BARUV(V,HGV,0) $ CALL UVCAL(ZM,-1.) $ CALL CONUV(V,HGV,0)
      DO 260 L=1,MAXLAY $ JL=LEV(L)+1 $ CALL XMIT(ZM(JL),V(JL),NO,1,1)
  260 CONTINUE
*BEFORE HNG8.118
      DIMENSION VU(2) $ ICON=0 $ GO TO 5 $ ENTRY CONUV $ ICON =1 $ CY=0.
    5 CONTINUE
*DELETE BARMOD.12,HNG8.125
   17 IF(ICON.EQ.1) GO TO 18
      ZM(JL)=B(JL)*ALPHA+(B(NM+JK)+B(NP+JK)+B(MM+JK)+B(MP+JK))*BETA
      GO TO 20
   18 VU(1)=VU(2)=1.
      IF((MM.EQ.I.OR.MP.EQ.I.AND.N.EQ.0).OR.(NM.EQ.I.OR.NP.EQ.I.AND.N.EQ
     *.-1)) VU(2)=2.
      IF((MOD(M+1,ME).LE.2.AND.N.EQ.-1).OR.(MOD(I+1,NE).LE.2.AND.N.EQ.0)
     *) VU(1)=0.
      VU(N+2)=(B(MP+JK)-B(MM+JK))*VU(N+2)
      VU(1-N)=(B(NM+JK)-B(NP+JK))*VU(1-N)
      CX=-A7*(B(JL)*VU(1)+ZST(JL)*VU(2)) $ ZM(JL)=ZM(JL)+CX
*IDENT CONVCU
*DELETE CONVCT.13,CONVCT.16
      IF(N.EQ.0) GO TO 181 $ IF(NM.EQ.I.OR.NP.EQ.I) VU(2)=2.
      IF(MOD(M+1,ME).LE.2) VU(1)=0. $ GO TO 182
  181 IF(MM.EQ.I.OR.MP.EQ.I) VU(2)=2. $ IF(MOD(I+1,NE).LE.2) VU(1)=0.
  182 CONTINUE
*IDENT CONPRT
*DELETE CONVCT.1
      B ,A7,CY,CZ(800)
*INSERT BC8.7
      IF(MOD(IT,IPMOD).EQ.0.AND.IT.GE.IPO) CALL PRTMAX(CZ((JL-1)*NO+1),
     *4HNONE,1H ,42HCONVECTIVE TERM IN U COMPUTATION FOR LAYER,1,42,IT)
```

```
*INSERT HNG8.54
      CALL PRTMAX(CZ((L-1)*NO+1),4HNONE.1H ,42HCONVECTIVE TERM IN V COMP
     *UTATION FOR LAYER,1.42.IT)
*DELETE HNG8.121
      J=(L-1)*NO+I $ IF(H(JL).GT.0.)GOTO 15 $ ZM(JL)=CZ(J)=0. $ GO TO 20
*INSERT CONVCT.19
      CZ(J)=CX $ CY=AMAX1(CY.ABS(CX))
*BEFORE HNG.246
  799 IF(CY.LT.1000.) GO TO 800 $ PRINT 1006.CY $ GO TO 801
*INSERT HNG.246
  801 CONTINUE
*DELETE UTIL.86
 2000 FORMAT(A1.I4,3X15E8.2)
*IDENT CONPRU
*DELETE CONPRT.1
     B .A7.CY.CZ(300)
*INSERT HNG8.20
      CALL XMIT(-0.,CZ(JL+1).MAXDIM,0,1)
*INSERT BC8.7
      IF(MOD(IT,MODOUT).EQ.0.AND.IT.GE.ITO)
     * CALL INPOUT(CZ(K-NO+1).JL,8,IT,CFAC)
*DELETE CONPRT.2.CONPRT.3
*INSERT HNG8.49
      CALL INPOUT(CZ(JL).L,8.IT,CFAC)
*DELETE CONPRT.4.CONPRT.5
*DELETE CONPRT.6
      IF(CY.GT.10.) PRINT 1006.CY
*DELETE CONPRT.7
*DELETE CONPRT.8
      IF(H(JL).GT.0.) GO TO 15 $ ZM(JL)=CZ(JL)=0. $ GO TO 20
*DELETE CONPRT.9
      CZ(JL)=CX $ CY=AMAX1(CY,ABS(CX))
*DELETE CONPRT.10
*RESTORE UTIL.86
*IDENT CONPRV
*DELETE CONPRU.4
     *CALL INPOUT(CZ(K-NO+1),JL,10,IT,CFAC)
*DELETE CONPRU.5
      CALL INPOUT(CZ(JL).L,10.IT,CFAC)
```

APPENDIX D

LAYER DISAPPEARANCE MODIFICATIONS

FLAT, FLATA AND FLATB SERIES

The FLAT, FLATA and FLATB modifications allow layer disappearance cases to be handled reasonably in the Phase II computation. The procedures maintain a closure status flag array based on a minimum allowed thickness, reset lesser thicknesses to the minimum thickness, and reset the old velocity components on closed sides to zero. For any closed side, the adjacent ζi heights are recomputed without smoothing using the modified thickness and old velocity fields. After the recomputation of ζi height, new velocity components are computed in the usual fashion.

The initialization section equivalences the flag field IZi to the ZSTi array, which is available at this phase of the computation for this special use. The LAYS card and interpretation section are modified to include a minimum thickness value, HMIN, in columns 71-75 and to calculate the constant HMIN4. SETHUV is modified to include an array index offset parameter so that array Zi is used in the computation for initialization only. Thereafter, the tidal and volume source specifications and thickness calculations are all modified to use ZMi (rather than Zi).

The thickness computation block is most heavily affected by the FLAT modifications. The bulk of the layer disappearance procedures follows the call to CXMIT that is used to set the Hvi boundary condition. The flag field, IZi, is initialized to display code zero (33B). For all sea points the average thickness must be greater than HMIN else the flat position is set to indicate cell closure (30 B). If the individual thickness of a cell side is less than HMIN, the flag field is incremented by an integer coded to show the particular side closed in the flag position except on full closure. The top side of a cell is indicated by an increment of 2, the right side by an increment of 8, the bottom by 4 and the left by 1. These integers are independent so the sides that are closed may be determined from the sum (TABLE D-1.) For each closed side the normal velocity component is set to zero and the thickness if less than HMIN is reset to HMIN. The boundaries of the flag field are adjusted based on the sequence of operation. Each cell is viewed only in terms of its north and west sides so that the computation does not progressively contaminate itself. Thc CXMIT routine is called to put the usual boundary condition back on the Ui, Vi, HGUi and HGVi arrays. This precaution must be taken since some of the recomputed ζi may be on the boundary. The flag field is used to control the recalculation of ζi. Note that recomputation cannot use the smoothed ζi since the ZM array is overwritten during the calculation of the required ζi.

The non-availability of the smoothed ζi has beneficial side effects: certain boundary configurations occur with the layer disappearance procedure which would require specialized computation of the smoothed component, but the limited testing of such specialized smoothing computations did not produce significantly different results so no smoothed component is included. Instead the old ζi is used from array Zi for the recomputation. The array ZMi is transmitted into array Zi after this computation by routine XMIT. The naming conventions for these modifications are similar to those described for CONVCT series, except that subsequent corrections must be named FLATA rather than FLAT to obtain the correct sequence ident.

TABLE D-1. TIDAL FLAT CODE CONVENTIONS

FLAG NO.	ASCII CHARACTER	OCTAL	CLOSURE STATUS
1	0	33	OPEN
2	1	34	W
3	2	35	N
4	3	36	NW
5	4	37	S
6	5	40	SW
7	6	41	SN
8	7	42	SNW
9	8	43	E
10	9	44	EW
11	+	45	EN
12	-	46	ENW
13	*	47	ES
14	/	50	ESW
15	(	51	ESN
16+	)	52	CLOSED (FLAT CELL)
17	X	30	CLOSED (FLAT CELL)
18++	L	14	CLOSED (LAND CELL)

+ Flag No 16 exists only temporarily before being replaced by Flag No 17. It represents a partially closed cell being fully closed by subsequent full closure of adjacent cells.

++ Flag No 18 exists only for printed display purposes.

FLATIO AND FLATIP SERIES

The FLATIO series of modifications provide an output capability to display the results of the tidal flat closure during the layer disappearnace procedures. The general technique has been to save the flag array on the intermediate data set, LU1, whenever a change of condition occurs in the particular layer. An attempt was made to provide multi-layer generality as in the layer disappearance procedure proper, but no testing of this function has been performed. A subroutine DISPLE with several entry points was designed to read, write and process the flag array as type 9 data records on LU1. Since the velocity fields and height fields are modified by the layer disappearance procedure, this procedure also implements the means of saving these modified arrays on the intermediate data set as type 3 data records at the usual tape save interval. At the conclusion of the run the intermediate data set file is rewound and the character flag fields read in and displayed. A call to the DSTART entry of DISPLE has been added to the initialization block to initialize the display package. The thickness computation block contains a call to routine DISPLE to process the flag field produced by the FLAT, FLATA series modification. A call to INPOUT is added to save the modified arrays at data save steps. The output block is expanded to incorporate a call to entry DISPRT of routine DISPLE, which rewinds LU1 and prints the flag fields saved during the course of the run.

Routine INPOUT is modified to handle the special type 9 data records, which unlike other arrays are packed as 6 bit display code characters rather than 24 bit integer quantities. The record also contains in the data portion just following the normal header two full word variables, the layer and the time. Special control is incorporated to avoid the 24 bit compression and scaling usually provided by INPOUT.

Routine DISPLE (Block N:FLATIO.4,66)

Routine DISPLE provides the means of saving the layer disappearance flag field as character arrays on the intermediate data set. The routine requires two parameters, JL and IT, which represent the layer and the time, respectively. Common/DLAY/ is introduced to save the time of last change, the layer and the last condition of the character flag array. A comment in the program indicates the required dimensionality for the members of common/DLAY. On the main entry the IZi array is converted to a display code character array and then compared to the old character array for the same layer. If they are the same, the routine is complete and returns to the main program; otherwise, the routine calls INPOUT for type 9 data record setting NO=1. When INPOUT is complete NO is reset to its normal value. The time level and the character array are moved into the space reserved for that level's old character array and return is made to LAYER3.

Entry DISPIN provides a restart capability, which has been tested and found to function properly. Its use requires special coding to recover the record desired, because the character arrays are saved at irregular times.

Entry DISPRT is called at the end of the main program after the time loop is complete. It provides the capability to read back in the saved type 9 records and restructures them into printable form using a DECODE statement to unpack the characters. One print line at a time is encoded and printed. The process cycles through all arrays and the program ends with the end of file error message printed by the IOERR routine called from INPOUT.

The actual form of the display of cell closure depends on the number of layers and the number of rows and columns in the defining mesh. If the required array size is sufficiently small, up to the number of layers plus one array displays may appear within the print line width. In any event at least one array will appear within the width of a print line, and this single array display is described.

The flag fields are printed in the form in which they are saved, i.e., as a series of two dimensional character arrays reflecting the changing character of the tidal flat region in change steps. The array includes the original land areas as the double character "L" (flag 18 of TABLE D-1), the closed tidal flat cells as double character "X" (flag 17) and the partially closed cells as single character flags indicating closure status (flags 1-16). Printed headings for each character array show the layer number for the array and the associated computation time in seconds during the course of the model run.

The DSTART entry computes constants required by the other calls to the routine so must be called first. Comments in the program define the constants.

LISTING FOR LAYER DISAPPEARANCE CORRECTION SETS

The following listing is the FLAT and FLATIO series correction sets in card image form as originally entered into the library. The spacing between correction sets is artificial and the sets are presented in a logical order rather than the order in which they were entered on the library. The descriptions rely on references to these listings and their relationship to the listing of the basic Phase II program presented in Figure 8. These correction sets reside on the library and the means of using these sets are presented in TABLE 11.

```
*IDENT FLAT
*INSERT HNG8.8
      *,(IZ,ZST) $ DIMENSION IZ(1)
*DELETE HNG8.14
   18 DECODE(76,1018,IC)HL,ALPHA,RHO,HMIN $ GO TO 1
*INSERT HNG8.23
   38 HMIN4=HMIN*4.
*DELETE HNG8.25
      CALL SETHUV(-3*MAXDIM)
*DELETE HNG8.32
      ZM(II)=ZM(II)-ZZ*A4 $ IF(JL.EQ.1)GO TO 50 $ JL=JL-1 $ GO TO 41
*DELETE HNG81.6
      K=HTZ(I) $ IF(K.GE.JL.AND.ZZ(JL).NE.0.) ZM(I+L)=ZZ(JL)
*DELETE HNG.134
  140 CALL SETHUV(0)
*INSERT BC8.6
      DO 142 JL=1,MAXLAY $ L=LEV(JL)+1 $ CALL XMIT(33B,IZ(L),NO,0,1)
  142 CONTINUE
      DO 151 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 151
      IMM=(M-1)*NE+1
      DO 150 I=IM,IP $ IF(HTZ(I).LE.0.)GO TO 150
      MM=I-NE $ IF(MM.LE.0)MM=I $ NM=MAX0(I-1,IMM) $ JL=HTZ(I)+1.
  143 ZZ=JL=JL-1 $ L=LEV(JL) $ II=L+I $ I1=L+MM $ I2=L+NM
      IF((HGU(II)+HGV(II)+HGU(I1)+HGV(I2)).LE.HMIN4) IZ(II)=30B
      IF(HTZ(NM).LT.ZZ) GO TO 146 $ IF(IZ(II).EQ.30B) GO TO 144
      IF(HGV(I2).GT.HMIN.AND.IZ(I2).NE.30B) GO TO 146
      IZ(II)=IZ(II)+2 $ IF(MOD(I,NE).EQ.0) IZ(II)=IZ(II)+4
  144 IF(IZ(I2).NE.30B) IZ(I2)=IZ(I2)+4
      HGV(I2)=AMAX1(HGV(I2),HMIN) $ V(I2)=0.
  146 IF(HTZ(MM).LT.ZZ) GO TO 148 $ IF(IZ(II).EQ.30B) GO TO 147
      IF(HGU(I1).GT.HMIN.AND.IZ(I1).NE.30B) GO TO 148
      IZ(II)=IZ(II)+1 $ IF(M.EQ.ME) IZ(II)=IZ(II)+8
  147 IF(IZ(I1).NE.30B) IZ(I1)=IZ(I1)+8
      HGU(I1)=AMAX1(HGU(I1),HMIN) $ U(I1)=0.
  148 IF(IZ(I1).EQ.52B) IZ(I1)=30B $ IF(M.NE.ME) GO TO 149
      IF(IZ(I2).EQ.52B) IZ(I2)=30B $ IF(I.NE.NO) GO TO 149
      IF(IZ(II).EQ.52B) IZ(II)=30B
  149 IF(JL.GT.1) GO TO 143
```

```
 150 CONTINUE
 151 CONTINUE
     DO 152 JL=1,MAXLAY $ K=LEV(JL)+NO
               CALL CXMIT(HGU(K-NE),HGU(K),HGU(K),0.,NE,-1,-1,-1)
               CALL CXMIT(HGV(K-1 ),HGV(K),HGV(K),0.,ME,-NE,-NE,-NE)
               CALL CXMIT( U(K-NE), U(K),HGU(K),0.,NE,-1,-1,-1)
               CALL CXMIT(V (K-1 ),V (K),HGV(K),0.,ME,-NE,-NE,-NE)
 152 CONTINUE
     DO 161 M=1,ME $ IM=NNM(M) $ IP=NNP(M) $ IF(IP.LT.IM) GO TO 161
     IMM=(M-1)*NE+1 $ DO 160 I=IM,IP $ IF(HTZ(I).LE.0.)GO TO 160
     MM=I-NE $ IF(MM.LE.0)MM=I $ NM=MAX0(I-1,IMM) $ JL=HTZ(I)$ ZZ=ZX=0.
 153 L=LEV(JL) $ II=L+I $ IF(IZ(II)-33B) 155,157,154
 154 I1=L+MM $ I2=L+NM $ IF(ZZ.EQ.0.) ZZ=ZX
     ZX=ZZ=ZZ-A4*(HGU(II)*U(II)-HGU(I1)*U(I1)+HGV(I2)*V(I2)-HGV(II)*
    *V(II))
 155 Z(II)=Z(II)+ZX $ ZX=Z(II)-ZM(II) $ ZM(II)=Z(II) $ GO TO 159
 157 ZM(II)=ZM(II)+ZX
 159 IF(JL.EQ.1)GO TO 160 $ JL=JL-1 $ GO TO 153
 160 CONTINUE
 161 CONTINUE
     DO 171 JL=1,MAXLAY $ L=LEV(JL)+1 $ CALL XMIT(ZM(L),Z(L),NO,1,1)
 171 CONTINUE


*DELETE HNG.249
      SUBROUTINE SETHUV(KZ)
*DELETE HNGB.66
      ZZ2=HL(JL)-(ZM(I1+KZ)+ZM(I2+KZ))*.5


*IDENT FLATA
*DELETE FLAT.19
  144 IF(I2.EQ.II) GO TO 146 $ IF(IZ(I2).NE.30B) IZ(I2)=IZ(I2)+4
*DELETE FLAT.21
  146 IF(HTZ(MM).LT.ZZ) GO TO 149 $ IF(IZ(II).EQ.30B) GO TO 147
*DELETE FLAT.24
  147 IF(I1.EQ.II) GO TO 149 $ IF(IZ(I1).NE.30B) IZ(I1)=IZ(I1)+8
```

```
*IDENT FLATIO
*INSERT HNG8.23
      CALL DSTART(L,ITS)
*INSERT FLAT.48
      CALL DISPLA(JL,IT)
*BEFORE HNG.247
      REWIND 1 $ CALL DISPRT(1,IT)
*INSERT BAPSTAR.50
      SUBROUTINE DISPLA(JL,IT)
*CALL CONTRL
*CALL BLANK
     *,IW(500)
      COMMON/DLAY/IFT(3),IFL(3),IFC(156)
C     DIMENSION IFT(MAXLAY),IFL(MAXLAY),IFC(KO*(NO.OF LAYERS+1))
      DIMENSION IZ(1),JZ(1)
      EQUIVALENCE (ZM(1),JZ(1)),(ZST(1),IZ(1))
      DATA (IB=1H )
      L=LEV(JL) $ K=1 $ KK=150
      DO 100 I=1,NO,150 $ IF(I+150.GT.NO) KK=NO-I+1
      IM=I+L $ IP=IM+KK-1
      ENCODE(KK,1000,IFC(K))(IZ(II),II=IM,IP) $ K=K+15
  100 CONTINUE $ L=JL*KO
 1000 FORMAT(150R1)
      DO 140 K=1,KO $ IF(IFC(K).NE.IFC(K+L)) GO TO 150
  140 CONTINUE $ RETURN
  150 CALL INPOUT(IFC(1),JL,9,IT,XX)
      CALL XMIT(IFC(1),IFC(L+1),KO,1,1)
      IFT(JL)=IT $ IFL(JL)=JL
      RETURN
C
      ENTRY DISPIN $ IFP=0 $ L=-JL $ K=JL*KO+1 $ GO TO 300
C
      ENTRY DISPRT
  190 IFP=1
  200 L=-IFP
  300 KK=IT $ CALL INPOUT(IFC(1),L,9,KK,XX) $ IF(IFP.EQ.0) GO TO 310
```

```
      IF(IFP.EQ.MZ) GO TO 350 $ K=IFP*KO+1
  310 CALL XMIT(IFC(1),IFC(K),KO,1,1) $ IFT(L)=KK $ IFL(L)=L
      IF(IFP.EQ.0) RETURN $ IFP=IFP+1 $ GO TO 200
  350 ZZ=JL=L $ IT=KK $ L=0 $ K=1 $ DO 361 KM=1,MZ $ IF(KM.EQ.1)GOTO 355
      K=KM-1 $ L=LEV(K)+MAXDIM $ ZZ=IFL(K) $ K=K*KO+1
  355 KK=150 $ DO 360 I=1,NO,150 $ IF(I+150.GT.NO) KK=NO-I+1
      IM=I+L $ IP=IM+KK-1
      DECODE(KK,1000,IFC(K)) (JZ(II),II=IM,IP) $ K=K+15
      DO 360 II=IM,IP $ J=II-L
      IF(JZ(II).GE.33B) JZ(II)=JZ(II)+5500B
      IF(HTZ(J).LT.ZZ) JZ(II)=1414B
      IF(JZ(II).EQ.5552B.OR.JZ(II).EQ.30B) JZ(II)=3030B
  360 CONTINUE
  361 CONTINUE
      DO 391 IM=1,NE $ IP=NO-NE+IM $ IF(MZ.GT.1) GO TO 385
      ENCODE(136,1383,IW(1)) I8,(JZ(I),I=IM,IP,NE)
 1383 FORMAT(68R2)
      IF(IM.EQ.1) PRINT 1384,JL,IT $ GO TO 390
 1384 FORMAT(1H0/1(7H LEVEL=I10,2X5HTIME=I10))
  385 ENCODE(136,1383,IW(1))(I8,(IZ(I+LEV(KM-1)),I=IM,IP,NE),KM=2,MZ),
     *                        I8,(JZ(I),I=IM,IP,NE)
      IF(IM.EQ.1) PRINT 1389,(IFL(KM-1),IFT(KM-1),KM=2,MZ),JL,IT
 1389 FORMAT(1H0/4(7H LEVEL=I10,2X5HTIME=I10))
  390 PRINT 1390,(IW(K),K=1,KZ)
 1390 FORMAT(13A10,A6)
  391 CONTINUE $ GO TO 190
C
      ENTRY DSTART $ KO=(NO+9)/10 $CALL XMIT(0,IFC(1),KO*(MAXLAY+1),0,1)
C        KO - NO OF WORDS PER CHARACTER ARRAY
C        NZ - NO OF CHARACTERS PER ROW IN PRINT LINE
C        MZ - NO OF ROWS PER PRINT LINE
C        KZ - NO OF WORDS PER PRINT LINE
      NZ=(ME+1)*2 $ MZ=MIN0(136/NZ,JL) $ KZ=(NZ*MZ+9)/10 $ XX=KO
      RETURN
      END
```

```
*INSERT UTIL.12
      M=FACOUT
*BEFORE UTIL.15
      IF(ITY.GT.7) GO TO 310
*BEFORE UTIL.34
  310 BUFFER IN(1,1) (IW(1),IW(LEN)) $ IREC=IREC+1
      IF(UNIT(1)) 350,320,330
  320 IERR=1 $ GO TO 340
  330 IERR=2
  340 PRINT 1130,IW(1),IW(2),ID1,ID2 $ CALL IOERR(IERR,IREC) $ GO TO 310
  350 ID2=SHIFT(IW(2),-48) $ IF(ID2.NE.9) GO TO 310
      CALL XMIT(IW(5),A(1),M,1,1) $ LAYER=IW(3) $IT=IW(4) $ GO TO 700
*INSERT UTIL.35
      IF(ITY.GT.7) GO TO 510
*DELETE UTIL.51
  500 CONTINUE    $ GO TO 600
  510 CALL XMIT(A(1),IW(5),M,1,1) $ NW=M+4 $ IW(3)=LAYER $ IW(4)=IT
  520 BUFFER OUT (1,1) (IW(1),IW(NW)) $ IREC=IREC+1
      IF(UNIT(1)) 700,530,540
  530 IERR=3 $ GO TO 550
  540 IERR=4
  550 CALL IOERR(IERR,IREC) $ GO TO 520
*INSERT UTIL.52
  700 CONTINUE


*IDENT FLATIP
*DELETE FLATIO.2
*INSERT FLAT.50
      CALL DISPLE(JL,IT)
      IF(MOD(IT,MODOUT).EQ.0.AND.IT.GE.ITO)CALLINPOUT(Z(L),JL,3,IT,CFAC)
*DELETE FLATIO.4
      SUBROUTINE DISPLE(JL,IT)
*DELETE FLATIO.21
  150 NO=1 $ CALL INPOUT(IFC(1),JL,9,IT,XX) $ NO=NE*ME
*DELETE FLATIO.68
*INSERT UTILA.3
```

```
      IF(ITY.EQ.9) GO TO 160
*INSERT UTIL.24
  160 ID2=SHIFT(IW(2),-48) $ IF(ID2.NE.9) GO TO 100
      CALL XMIT(IW(5),A(1),M,1,1) $ LAYER=IW(3) $ IT=IW(4) $ GO TO 600
*DELETE FLATIO.69,FLATIO.75
*DELETE FLATIO.76
*BEFORE UTIL.37
      IF(ITY.EQ.9) GO TO 405
*INSERT UTIL.37
  405 CALL XMIT(A(1),IW(5),M,1,1) $NW=M+4 $ IW(3)=LAYER $ IW(4)=IT
      GO TO 475
*RESTORE UTIL.51
*DELETE FLATIO.77,FLATIO.83
*DELETE FLATIO.84
```

APPENDIX E

MONTE CARLO MODIFICATION

MONTE

The Monte Carlo modification uses a pseudo-random number generator to effect diffusion in an advected field of lagrangian marker points. HN Phase II generated velocities are interpolated to the prior position of each marker point, then modified by a random increment to determine the new position. The version on the HN library now allows up to 1000 marker points, which are displayed through time to show the dispersion of each source array. In the present code a single marker point is introduced at each time step simulating a continuous source. The procedure operates as a part of Phase II and the printed output feature is an integral part of the modification rather than a separate option as in the convective and tidal flat extensions. The marker point position data are not saved on the intermediate data set although a data saving feature would fit within the present framework of data record types. The two forms of printed output generated are described as the modification to the Phase II blocks and additional routines are presented.

The initialization block contains the 1000 position arrays, XS and YS, that are inserted in control common immediately before the 500 word subroutine scratch area and are used to keep track of the present position of each marker point. The number of points followed depends on the number of SORC cards and the number of time steps. If the number of SORC cards times the number of time steps exceeds 1000 the XS and YS dimensions must be expanded. The SORC card used with the Monte Carlo extension was modified to include 2 additional floating point values per card in order to specify the exact source point location. This change required modification of array SORC and the DECODE processing for the SORC card. The ζ computation block contains the initialization of the position array, but the bulk of the Monte Carlo procedure follows the thickness computation and is considered part of the thickness computational block. When the layer disappearance procedure is included the Monte Carlo procedure follows the tidal flat procedure. Each marker point considered for a particular step is advanced by modification of the position arrays in accordance with the interpolated and randomly incremented velocity and the time step. The function WAYTUV does a simple bi-linear interpolation between the nearest four like velocity components. Information is retained during the position computation to provide a print line of statistics for each source at each time step giving the x-y components of center of mass, distance to center of mass from the source, horizontal width of the mass and standard deviation of the marker points about the center of mass.

In addition to the statistical information, a second printout is produced for each source at each print step showing the point positions as a two dimensional character array in which empty rows are deleted. Each printed array presents the essential portion of a variable dimension square window view of the point distribution for each source card. Each element of the 50 row by 100 column array represents one five-thousandth of the entire area of the square window. The dimensions of the window are determined by the extent of the point distribution. The characters printed each represent a count of the marker points falling within each sub-window array position at the print time. Total row counts appear to the right of the printed output and total column counts appear in an offset array format with each count right-justified immediately below its respective column. Empty rows are not printed so a row identification of absolute floating N index and relative position within the window is printed on the left. Similar absolute M index and window relative information are provided above the columns. The absolute and relative information refer to the upper left hand corner of the print position. The output block prints the final position of the random number generator for use in restarting the model.

Function WAYTUV (Block O:MONTE.64,76)

The function WAYTUV returns the value of the interpolated velocity component based on the input parameter list. The first parameter is the leftmost index in the x direction, the second parameter is the uppermost index in the y direction, the third parameter specifies an offset which determines whether a u or v computation will be made. The routine computes the bracketing indices around the x (or y) position, then computes linearly interpolated values in the x direction and finally linearly interpolates the resulting values in the y direction.

LISTING FOR MOTE CARLO CORRECTION SETS

The following pages present the MONTE correction sets in card image form as originally entered into the library as modifications to Phase II. These correction sets reside on the library and the means of using these sets are presented in TABLE 11.

```
*IDENT MONTE
*INSERT HNG8.1
     * ,XS(1000),YS(1000)
*DELETE HNG8.5
     1,TIDSPD(4),IFLOW(8,30),FLOW(4,30),ISORC(8,10),SORC(4,10),
*INSERT HNG.15
     * ,(NPOC=61)
*INSERT HNG8.11
      POC=NPOC-1
*DELETE HNG.44
      DECODE(76,1020,IC)(ISORC(I,ISNO),I=1,8),(SORC(I,ISNO),I=1,4)
*INSERT HNG.120
      JL=0 $ DO 81 K=1,ISNO $ NPTS=SORC(1,K)
       IF(IT.NE.ISORC(7,K)) GO TO 81
       XS0=SORC(3,K) $ YS0=SORC(4,K)
      DO 80 II=1,NPTS $ I=II+JL $ XS(I)=XS0 $ YS(I)=YS0
   80 CONTINUE
   81 JL=JL+NPTS
*INSERT FLAT.51
      IF(ISNO.EQ.0) GO TO 199 $ JL=0
      DO 198 K=1,ISNO $ NPTS=(IT-ISORC(7,K))/ITINC+1
      IF(IT.LT.ISORC(7,K).OR.IT.GT.ISORC(8,K)) GO TO 198
      PC=1./NPTS $ XS0=SORC(3,K) $ YS0=SORC(4,K)
      XMIN=YMIN=MAXDIM         $ XMAX=YMAX=XSUM=YSUM=0.
      DO 180 II=1,NPTS $ I=II+JL
      UR=RANF(FTN)-.5                   $ VR=RANF(FTN)-.5
      USX=WAYTUV(XS(I)-.5,YS(I),0)+UR
      VSX=WAYTUV(XS(I),YS(I)-.5,MAXDIM)+VR
      IF(UR.NE.USX) XS(I)=USX*A4+XS(I) $ XSUM=XS(I)+XSUM
      IF(VR.NE.VSX) YS(I)=-VSX*A4+YS(I) $ YSUM=YS(I)+YSUM
      XMAX=AMAX1(XS(I),XMAX) $ XMIN=AMIN1(XS(I),XMIN)
      YMAX=AMAX1(YS(I),YMAX) $ YMIN=AMIN1(YS(I),YMIN)
  180 CONTINUE $ MP=XMAX+1. $ NP=YMAX+1. $ MM=XMIN $ NM=YMIN
      M=W=AMAX1(XMAX-XMIN,YMAX-YMIN) $ IF(M.EQ.0) M=W=1.
      AXS=XSUM*PC $ XMAX=XMAX-AXS $ AYS=YSUM*PC $ YMAX=YMAX-AYS
      SIG2X=SIG2Y=0. $ DO 182 II=1,NPTS $ I=II+JL
      SS=XS(I)-AXS $ SIG2X=SS*SS+SIG2X $ SS=YS(I)-AYS $SIG2Y=SS*SS+SIG2Y
  182 CONTINUE $ SIG2X=PC*SIG2X $ SIG2Y=PC*SIG2Y
      AXS=AXS-XS0 $ AYS=AYS-YS0
      RVEC=SQRT(AXS*AXS+AYS*AYS)
      RANG=AMOD(ROT-ATAN2(AYS,AXS)*57.295779513,360.)
      PRINT 1180,NPTS,NM,NP,MM,MP,XMIN,XMAX,YMIN,YMAX,SIG2X,SIG2Y
     * ,RVEC,RANG,AXS,AYS,UR,VR
 1180 FORMAT(I5,8H POINTS,4I3,4H, X=2F7.3,4H, Y=2F7.3,5H, S2=2F7.3,
     *4H, S=F7.3,1H,F5.1,4H, A=2F7.3,4H, P=2F6.4)
      IF(MOD(IT,IPMOD).NE.0) GO TO 198
      PC=50./W $ NY=XMIN*1000. $ MX=(XMIN+W)*1000.
      DO 188 J=1,51 $ IW(J)=MOD(2*J-2,10)+5533B
  188 CONTINUE $ WK=W*DL*1.E-5
      PRINT 1188,K,W,WK,IT,NY,MX,(IW(J),J=1,51) $ I2=0 $ W=W*10.
 1188 FORMAT(1H020X17HWINDOW FOR SOURCEI3,3H ISF8.3,12H GRID CELLS(F8.3,
     *18HKM) SQUARE AT TIMEI10/11XI6,I96/11X51R2)
      CALL XMIT(0,IW(104),101,0,1) $ CALL XMIT(55B,IW,103,0,1)
      DO 197 N=1,51 $ I1=0
      DO 192 II=1,NPTS $ I=II+JL
      NY=(YS(I)-YMIN)*PC+1. $ IF(NY.NE.N) GO TO 192
      MX=(XS(I)-XMIN)*PC*2.+3. $ IW(MX)=IW(MX)+1 $ I1=I1+1
  192 CONTINUE $ IF(I1.EQ.0) GO TO 197 $ I2=I2+I1
      DO 196 J=3,103 $ L=IW(J)-56B $ IF(L)196,195,194
  194 IW(J)=L+34B $ IW(J+101)=IW(J+101)+L+1 $ GO TO 196
  195 IW(J)=IW(J)+1 $ IW(J+101)=IW(J+101)+1
  196 CONTINUE
      IW(2)=2*N-2 $ IW=YMIN*1000.+FLOAT(IW(2))*W $ IW(2)=MOD(IW(2),100)
      PRINT 1196,(IW(I),I=1,103),I1 $ CALL XMIT(55B,IW,103,0,1)
 1196 FORMAT(I8,I3,1X101R1,I7)
  197 CONTINUE $ PRINT 1197,IW(204),((IW(J+I-1),J=104,199,5),I=1,5),I2
 1197 FORMAT(I8,20I5/9X20I5/10X20I5/11X20I5/12X20I5,I8)
  198 JL=JL+NPTS
  199 CONTINUE
*BEFORE HNG,247
      CALL RANGET(X) $ PRINT 1800,X
 1800 FORMAT(24H1RANDOM SEED FOR RESTARTO22)
```

```
*INSERT BARSTAR.50
      FUNCTION WAYTUV(X,Y,NOFF)
*CALL CONTRL
*CALL BLANK
      X1=M=X $ X1=X-X1 $ X2=1.-X1
      Y1=N=Y $ Y1=Y-Y1
      NP=MIN0(N+1,NE) $ N=MAX0(N,1)
      MP=MIN0(M+1,ME) $ M=MAX0(M,1)
      M=(M-1)*NE+NOFF $ MP=(MP-1)*NE+NOFF
      U1=U(N+M)*X2+U(N+MP)*X1
      U2=U(NP+M)*X2+U(NP+MP)*X1
      WAYTUV=(1.-Y1)*U1+Y1*U2
      RETURN
      END
```

APPENDIX F

PHASE IV PROGRAM SERIES

THERMAL ADVECTION PROGRAM

Phase IV consists of the thermal advection routines which were built from a version of the Pederson-Prahm advection by method of moments and a thermal treatment heat budget package originated by T. Laevastu. A major difference between this and other phases of the HN system is the means of storing array elements by sequential column positions within sequential rows. Phases I and II and the intermediate data set use the opposite convention of sequential row positions within sequential columns. This change of convention requires modification to the I/O routine indexing both on input to include conversion to the internal storage convention and on output to provide conversion back to the convention defined by the intermediate data set.

Program ADVECT (Blocks A-E:A2.1,A1.27)

The program ADVECT consists of blocks A-E as identified in Figure 6. This routine acts as the control program not only for the advection by method of moments but also to control the thermal treatment of the advected excess heat quantity. Logical units are specified for the thermal advection routines as follows:

Logical Unit	Description
INPUT	Punched card input file
OUTPUT	Printed output file
1 (TAPE1)	Intermediate data set file
2 (TAPE2)	Thermal advection data set file
3(TAPE3)	≡ OUTPUT

The blocks A-E described below are arbitrary modules of the program chosen for convenience in description.

Advect Initialization (Block A:A2.1,36)

The program card specifies the logical units. Common /A/ includes general control and some specific control parameter for the various subroutines; blank common holds the major array variables associated with the advection procedure and the heat/temperature/temperature anomaly array, T; common/1/ provides space required by array BDVECT for the advection by method of moments; common/ABC/ provides the major arrays associated with the thermal process except the temperature array; and common/THERM/ provides central values to the thermal process. Dimensionality and equivalence classes are established for local routine use. Various constants are defined including some repetitive definitions due to temporary modifications. The modification TA800, for example, provides (when present) the capability to handle a 20 by 40 array. Without this modification only 300 position arrays of fewer than 30 rows are allowed. The master control cards are read in character mode and immediately printed for verification. Control then transfers to processing the specific card identified from the table of allowable card names, NAME. The position of the card names in this array is used as an index for routing control as in the other phases. Some default values are provided where no value is specified. Too many wind cards will not cause the routine to abort, but an informative diagnostic is provided.

If ME is greater than or equal to IDIM an information diagnostic is printed before early termination. Too many SORC cards cause immediate termination without diagnostic. Unidentified cards are ignored to avoid having to provide detailed recognition of cards not used in this phase: FACT, TIDE, FLOW, LAYS.

All accepted cards return to process the next card except the COMP card, which causes the execution of the rest of the routine. Additional constants are defined and the first and second floating values from each SORC card are multiplied by the time step to allow the deviant usage of volume and concentration (heat) per second rather than per step. Arrays are initialized by calls to routine XMIT, and a restart procedure is invoked when ITS is not equal to zero. The restart modification should be present only when the program is being started from a previous thermal advection data set. This modification causes type 20 data records to be read from LU2.

Thermal Treatment (Block B:A3.4,9)

From the beginning of this program block to the end of the output block the processing occurs for each time step specified on the TIME card. The first time through the loop the thermal treatment block is partially bypassed to avoid advection or thermal treatment in the first step. For all subsequent steps the following procedures occur: the HN Phase II velocity fields are read from the intermediate data set using INPOUT. If this is a print step, the velocity field is printed by PRTMAX, the thermal treatment occurs in routine THERMAL, and the output cell temperatures calculated in THERMAL are converted to a temperature anomaly by interpolating the width factors produced in the last advection step. For all steps including the first step the excess heat field is computed from the temperature anomaly field. Then, for the first step only, a bypass occurs to avoid the duplex advection procedure. Zero values in most arrays represent ambient values rather than true zero values. This feature restricts the field for computations and printed output to essential values required for computation of non-ambient cases.

Duplex Advection (Block C:A2.47,52)

Block C processing commences with three calls to routine XMIT. COMMON/1/ is used as temporary storage to swap the Q and V arrays, minimizing the changes required in routine BDVECT. Routine BDVECT does the minimal advection computation for the excess heat component (zeroth moment only.) On return XMIT is called again to reverse the swap. BDVECT is called again to advect the V array containing variable V applying the full method of moments using the same velocity field as in the first call for heat advection. This duplex advection block is bypassed for the first step. In the restart cases this prevents duplication of the last step of the prior run. In the normal start the bypass avoids advection of no quantity as the sources are applied subsequent to the advection step.

Sources; Temperature Anomaly; Wind (Block D:A2.53,78)

At and after the time the earliest source is specified to start, all source specifications are reviewed to determine whether they apply to the current step. When a source is relevant it is applied. The start time of the earliest source should correspond to the start time of the run on the TIME card. If either center of mass indicator computed by BDVECT exceeds .5 in absolute value at any point in the field, a warning indicating that the time step for the process is too long is printed and the routine is terminated. The center of mass is computed relative to the center of the cell in grid step units. The advected temperature anomaly is computed from the advected excess heat and the advected volume. Routine XMIT zeros the wind field, then the wind specifications are applied such that any wind scheduled to start at or prior to the current step actually begins to have effect in the subsequent step and carries its effect through to the indicated end time. A wind intended to begin prior to the end of the presently active specified wind will not take effect until the active wind stops.

Wind cards must be in order of start times and their start times and end times must be precise multiples of the run time step. Multiple wind subfields scheduled to start at the same time will take effect at the same time with the most recent specifications determining in overlap cases.

Output Processing (Block E:A2.79,A1.27)

Data save steps are defined by the initial output time ITO and the interval MODOUT from the TIME card. Routine INPOUT is called to place data record type 20 on the thermal advection data set file, LU2, at each data save step. Print steps are defined by the initial printed output time IPO and the modular interval IPMOD from the time card. Routine PRTMAX is called to print several arrays. If the current time exceeds the end time for the run, TE, the job terminates; otherwise, the program loops for the next time step.

Routine PRTMAX (Block F:A1.28,61)

The major differences between PRTMAX and the routine of the same name in Phase II are: for most arrays in the thermal advection, zero values are carried in lieu of ambient values; rows consisting entirely of zeroes are not printed; the level parameter is missing; and the method of indexing. Since the internal storage conventions differ, the indexing to achieve printed output is also different.

Routine THERMAL (Block G:T1.9,TH.324)

All required constants and variables are either local or in common. Default values are provided for all constants, though for a particular case these should be redefined to reflect the specific conditions desired. In the current version, the ambient water temperature is assumed constant as is the effect of solar heating. (The imposition of a diurnal cycle is such an individual phenomenon it is left to be determined on a case by case basis.)

First the heated fluid temperature anomaly is interpolated to the cell temperature based on the distribution of the fluid within the cell and the ambient water temperature and the saturated and moist water vapor pressures are defined for affected cells based on water temperature and relative humidity.

Effective back radiation (the net of oceanic and atmospheric longwave radiation), convective transfer of sensible heat (conduction from the ocean surface to the air molecules), and transfer of energy by evaporation (formation of atmospheric latent heat) or condensation (recovery of heat from atmosphere) are the heat budget components computed. The heat budget components and the volume associated with the heated fluid determine the temperature decrement for the step. If the cell temperature is adjusted to less than or equal to the ambient temperature, the volume, temperature and associated center of mass and width factors are set to zero. Several arrays are then printed by PRTMAX. All of the arrays are printed in restricted format due to the convention of maintaining zero rather than ambient values for unaffected cells or cells not required in a computation. Return to the main program is made with the array T≡Q holding the value of the temperature in the cell.

Routine BDVECT (Block H:A2.97,A3.26)

A single parameter is required by routine BDVECT to distinguish between calls for advection of the excess heat and for advection of the associated fluid volume. The parameter is used specifically to bypass first and second order moment computations for the heat advection, and to tailor the initialization processes for the difference in requirements. COMMON/1/ is used to hold seven intermediate accumulation computations by row. Each accumulation array holds space for present, prior and succeeding row position accumulations plus space for adjacent outside column position accumulation. The computations are done row by row with the accumulation arrays cycled to the next row setup after transfer of gross accumulation for prior row into the advection array. During the cycling process, the accumulator positions for the new succeeding row are set to 0. Quantities advected into the zeroth row or column, the ME+1 column or the NE+1 row are lost. The direction of the flow field restricts computation to the four or fewer accumulation cells: the source cell, a diagonal cell and two adjacent cells forming a square of side 2*DL. The advection subroutine code has only been modified to the extent required to handle the dual problem. Much could be done to improve the speed if heavy use is anticipated.

Routine XMIT (Block I:A1.62,A1.65)

See Appendix B for description of routine XMIT.

Routines INPOUT, IOERR (Block J:A1.81,A .144)

Routine INPOUT for Phase IV is similar to the routine of the same name used for Phases I and II from deck UTIL. Ultimately the same routine could be used for all four phases with simple modifications applied for the special requirements of each phase. The three major changes made to the Phase IV version are: 1) when the velocity components are read from the intermediate data set as type 2 or type 3 data records, the Ui values overlay the ζi values which are not required (to avoid future incompatibility this function was set to occur when ITY<0; 2) when the data are transferred from and to internal arrays the indexing is changed so internal array storage has the column index varying faster; and 3) six arrays are processed in a single call to INPOUT using LU2 when type 20 is specified.

Routine IOERR has been modified to identify the logical unit in the error message through an additional formal parameter in the call sequence.

LISTING FOR PHASE IV SAMPLE RUN

Following is a reproduction of a Phase IV run. The first page reflects the UPDATE processing of the library according to the library directives set input on cards with listing option L=124 as shown in the SCOPE control card example of TABLE 8. The FORTRAN compiler listing of Phase IV shows the UPDATE module identification and card number to the right. The run results are presented immediately following the listing of routine THERMAL. The input master control cards are printed directly as input followed by a series of arrays printed with empty lines deleted. The INPOUT lines appear as the type 20 restart fields are saved on LU2. The format for array print cannot print negative numbers except as a row of asterisks. This phenomenon may occur in the heat of evaporation field, indicating a net gain due to condensation at the surface, and in the center of mass component fields, indicating negative direction to center of mass.

```
CARDS ENCOUNTERED IN INPUT                          BKY 76/66 UPDATE 1.2/410-1H   09 MAR 77 01.15.35       PAGE   1
*****    *COMPILE AD,TH

         MODIFICATIONS / DIRECTIVES

YANK$$$  *YANK VARWINE                                                    AVARW2        1
YANK$$$  *YANK CONPRV                                                     ACONV2        1
YANK$$$  *YANK HNG7                                                       AHNG71        1
YANK$$$  *YANK VARCORL                                                    AVARC1        1
YANK$$$  *YANK VARWIND                                                    AVARW1        1
YANK$$$  *YANK VALDEZ                                                     AVALDEZ       1
YANK$$$  *YANK CONVCT,CONVCU                                              ACONV1        1
YANK$$$  *YANK CONPRT,CONPRU                                              ACONV1        2
YANK$$$  *YANK FLAT,FLATA                                                 AFLAT1        1
YANK$$$  *YANK FLATIO,FLATIP                                              AFLAT1        2
YANK$$$  *YANK MONTE                                                      AMONT1        1
YANK$$$  *YANK HNJ                                                        AHNIJ1        1
YANK$$$  *YANK BKYPRT                                                     AHNIJ1        2
AD       *CALL A                                                          A1            3
AD       *CALL B                                                          A1            4
AD       *CALL ABC                                                        A2            2
AD       *CALL THERM                                                      A2            3
AD       *CALL A                                                          A1           30
AD       *CALL A                                                          A1           66
AD       *CALL B                                                          A1           67
AD       *CALL C                                                          A1           68
AD       *CALL A                                                          A1           82
TH       *CALL A                                                          T2            1
TH       *CALL B                                                          T2            2
TH       *CALL ABC                                                        T2            3
TH       *CALL THERM                                                      T2            4
```

```
      PROGRAM ADVECT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3=OUTPUT)                 A2.1
C     CONVERTED TO CDC 7600 BY A.D.STROUD.CSI  -  MARCH 1976                A1.2
C     ROUTINE BASED ON J. HARDING PROGRAM FEDBACK                           AD.3
C     MODIFIED BY R. BAUERCOMPASS SYSTEMS, INC.    FEB.75                   AD.4
C     PROGRAMMED FOR CDC1604 UNDER CSIROS                                   AD.5
      COMMON/A/ ME,NE,ROT,DL,MAXDIM,MODEL,IDIM,NO,MINX,MINY,MAXX,MAXY,DTA.2
     *,FRACTX,FRACTY,IDIM4,NXP1                                             A.3
     *,NEM,TEM,TE,TWO                                                       COM.1
      COMMON CT(300),FXT(300),FYT(300),RXT(300),RYT(300)                    COM.2
     *,T(300),VX(300),VY(300)                                               COM.3
      COMMON/1/IW(960)                                                      A1.5
      COMMON/ABC/TA(300),EW(300),EA(300),XK(300),YK(300)                    COM.5
     * ,QB(300),QH(300),QE(300)                                             COM.6
      COMMON/THERM/IT,ITS,IPMOD,TL,CC,UU,K,C,EC,EK,KI,EKI,Q2T,DD,R          COM.7
     *,IPO                                                                  T3.28
      REAL K,KI                                                             THERM.6
      DIMENSION NAME(6),IC1(9),IC(1),ISORC(8,8),SORC(4,8),Q(1),V(1)         A2.4
     * ,IWIND(8,4),WIND(4,4)                                                A2.5
      EQUIVALENCE(IC1(2),IC(1))                                             A1.8
     *,(Q,T),(V,CT)                                                         A2.6
      DATA(NAME=4HNAME,4HGRID,4HTIME,4HSORC,4HWIND,4HCOMP),(IWNO=0)         A2.7
     1,(MAXDIM=300),(IDIM=30),(MAXNAME=6),(MAXSORC=8),(MAXWIND=4)           A2.8
     2,(MINSORC=99999999),(MINWIND=99999999),(MINX=1),(MINY=1),(ISNO=0)     A2.9
     3,(IC=5H(1H1)),(F=1000.),(TWO=10.),(TE=1.E-10),(CB=.956),(RO=1.025)A2.10
     *,(RPD=.01745329252)                                                   A2.11
C     UNITS---- CGS SYSTEM WITH CALORIES(GRAM-CALORIES)                     A2.12
      ROCB=RO*CB                                                            A2.13
      PRINT IC                                                              A2.14
C     READ CONTROL CARDS                                                    AD.17
    1 READ 1000,IC1                                                         A1.10
    2 PRINT1001,IC1 $ DO 3 I=1,MAXNAME                                      AD.19
 1000 FORMAT(A4,7A10,A6)                                                    A1.11
 1001 FORMAT(1XA4,7A10,A6)                                                  A1.12
      IF(IC1(1).EQ.NAME(I)) GO TO 4                                         AD.22
    3 CONTINUE $ GO TO 1                                                    AD.23
    4 GO TO (1,5,6,7,8,10),I                                                A2.15
    5 DECODE(76,1004,IC)NE,ME,DL,ROTANG,C3 $ ROT=810.-ROTANG                A2.16
      IF(ME.LT.IDIM) GO TO 1 $ PRINT 9005 $ GO TO 999                       A3.1
 9005 FORMAT(92H IDIM MUST.GT.ME, COMMON/1/IW(7*IDIM4),BDVECT COMMON/1/CA3.2
     *ACC(IDIM4),...,C(MAX0(1,NO-7*IDIM4)))                                 A3.3
 1004 FORMAT(2I3,7E10.3)                                                    AD.27
    6 DECODE(76,1005,IC)MODOUT,ITO,ITS,ITE,ITINC,IPO,IPMOD                  AD.28
      MODOUT=MAX0(ITINC,MODOUT,1) $ IPMOD=MAX0(IPMOD,ITINC,1)               A1.13
      GO TO 1                                                               AD.30
 1005 FORMAT(I6,7I10)                                                       AD.31
    7 ISNO=ISNO+1 $ IF(ISNO.GT.MAXSORC) GO TO 999                           AD.32
      DECODE(76,1007,IC)(ISORC(I,ISNO),I=1,8),(SORC(I,ISNO),I=1,4)          A2.17
 1007 FORMAT(4I3,2I2,2I10,4E10.3)                                           A2.18
      MINSORC=MIN0(MINSORC,ISORC(7,ISNO)) $ GO TO 1                         A1.14
    8 IF(IWNO.LT.MAXWIND) GO TO 9 $ PRINT 1008 $ GO TO 1                    A2.19
 1008 FORMAT(24H0***UNABLE TO PROCESS***/)                                  A2.20
    9 IWNO=IWNO+1                                                           A2.21
      DECODE(76,1007,IC)(IWIND(I,IWNO),I=1,8),(WIND(I,IWNO),I=1,4)          A2.22
      GO TO 1                                                               A2.23
   10 DECODE(76,1005,IC) MODEL,IFIL,LAYER $ NEM=NE-1 $ TEM=1.-TE            A2.24
      IF(LAYER.EQ.0) LAYER=1                                                A1.15
      DT=ITINC $ MAXX=ME $ MAXY=ME $ NO=NE*ME $ NXP1=ME+1                   A1.16
      FRACTX=FRACTY=DT/DL $ IDIM4=IDIM*3                                A3.4
      Q2T=DT*DL*DL/(24.*3600.*ROCB)                                     A2.25
      DO 19 J=1,ISNO $ SORC(1,J)=SORC(1,J)*DT $ SORC(2,J)=SORC(2,J)*DT  A2.26
   19 CONTINUE                                                          A2.27
      TL=FRACTX $ EKI=1./EK $ KI=1./K                                   A2.28
      C3=C3*DT*10000. $ MAXWIND=IWNO $ IWNO=0 $ ITW=ITS                 A2.29
      CALL XMIT(TWO,TA,NO,0,1)                                          A2.30
      CALL XMIT(0.,CT,8*MAXDIM,0,1)                                     A2.31
      CALL XMIT(0.,EA,6*MAXDIM,0,1)                                     A2.32
      DO 800 IT=ITS,ITE,ITINC                                           AD.39
      IF(IT.EQ.ITS) GO TO 50                                            A3.5
      CALL INPOUT(VX,-LAYER,-2,IT,F)                                    A3.6
      IF(IT.GE.IPO.AND.MOD(IT,IPMOD).EQ.0)                              A2.35
     *CALL PRTMAX(VY,VX,1H ,8HVELOCITY,8,IT)                            A2.36
      CALL THERMAL                                                      A2.42
      DO 30 I=1,NO $ P=RXT(I)*RYT(I) $ IF(P.GT.TE) T(I)=(T(I)-TWO)/P    A3.7
C     INTERPOLATE TO TEMP ANOMALY OF HEATED FLUID FROM CELL TEMPERATURE A3.8
   30 CONTINUE                                                          A2.46
   50 DO 60 I=1,NO $ IF(V(I).GT.TE) Q(I)=T(I)*V(I)*ROCB                 A3.9
C     COMPUTE EXCESS HEAT IN FLUID BASED ON ANOMALOUS TEMPERATURE       A3.10
   60 CONTINUE $ IF(IT.EQ.ITS) GO TO 70                                 A3.11
      CALLXMIT(Q,IW,NO,1,1)$CALLXMIT(V,Q,NO,1,1)$CALLXMIT(IW,V,NO,1,1)  A2.47
      CALL BDVECT(1)                                                    A2.49
      CALLXMIT(V,IW,NO,1,1)$CALLXMIT(Q,V,NO,1,1)$CALLXMIT(IW,Q,NO,1,1)  A2.50
      CALL BDVECT(7)                                                    A2.52
   70 IF(IT.LT.MINSORC)GO TO 80 $ DO 79 J=1,ISNO                        A2.53
      IF(ISORC(7,J).GT.IT.OR.ISORC(8,J).LT.IT)GOTO79                    A2.54
      LM=ISORC(5,J) $ LP=ISORC(6,J)$IF(LAYER.LT.LM.OR.LAYER.GT.LP)GOTO79AD.42
      NM=ISORC(1,J) $ NP=ISORC(2,J)                                     AD.43
      DO 78 N=NM,NP $ IP=(N-1)*ME $ IM=IP+ISORC(3,J) $ IP=IP+ISORC(4,J) AD.44
      DO 77 I=IM,IP                                                     AD.45
C     INSERT LOGIC FROM ROUTINE MERJE                                   AD.46
      IF(ABS(FXT(I)).LE..5.AND.ABS(FYT(I)).LE..5) GO TO 74              A1.18
      PRINT 7000,IT,I,FXT(I),FYT(I) $ GO TO 999                         AD.48
 7000 FORMAT(≠ POSSIBLE ADVECTION ERROR, TIME STEP TOO LONG≠,2I8,2E10.3)AD.49
   74 SUM=V(I)+SORC(2,J)                                                A2.56
      Q(I)=SUMQ=Q(I)+SORC(1,J) $ IF(SUM.GE.TE) GO TO 75                 A2.58
      CT(I)=SUM $ RXT(I)=RYT(I)=FXT(I)=FYT(I)=0. $ GO TO 77             AD.51
   75 CONCSM=V(I)/SUM $ V(I)=SUM                                        A2.59
      FXT(I)=FXT(I)*CONCSM $ FYT(I)=FYT(I)*CONCSM                       A1.21
      RXT(I)=1.-2.*ABS(FXT(I)) $ RYT(I)=1.-2.*ABS(FYT(I))               A1.22
   77 CONTINUE                                                          AD.56
   78 CONTINUE                                                          AD.57
   79 CONTINUE                                                          AD.58
      IF(IT.GE.IPO.AND.MOD(IT,IPMOD).EQ.0)                              A3.12
     * CALL PRTMAX(Q,4HNONE,1H ,20HADVECTED EXCESS HEAT,20,IT)          A3.13
   80 DO 89 I=1,NO $ IF(V(I).GT.TE)GO TO 88 $ T(I)=0. $ GO TO 89        A2.60
   88 T(I)=Q(I)/(V(I)*ROCB)                                             A2.62
   89 CONTINUE                                                          A2.63
   90 IF(IT.NE.ITW) GO TO 140 $ IF(IWNO.EQ.0) GO TO 91                  A2.64
      IF(IT.EQ.IWIND(7,IWNO)) GO TO 120                                 A2.65
   91 CALL XMIT(0.,XK,MAXDIM*2,0,1) $ ITW=0                             A2.66
  100 IWNO=IWNO+1 $ IF(IWNO.GT.MAXWIND) GO TO 140                       A2.67
      IF(IT.GE.IWIND(6,IWNO)) GO TO 100                                 A2.68
      ITW=IWIND(7,IWNO) $ IF(ITW.GT.IT) GO TO 140                       A2.69
  120 YW=(-ROTANG-90.-WIND(1,IWNO))*RPD                                 A2.70
      TK=WIND(2,IWNO) $ TK=TK*TK*C3 $ XW=COS(YW)*TK $ YW=SIN(YW)*TK     A2.71
      PROGRAM ADVECT   7600→7600 OPT=2                          FTN 4.6+439/040    09 MAR 77 01.15.36$ BKY PAGE      3

                      NM=IWIND(1,IWNO) $ NP=IWIND(2,IWNO)                              A2.72
                      DO 130 N=NM,NP $ IP=(N-1)*ME $ IM=IP+IWIND(3,IWNO)               A2.73
                      IP=IP+IWIND(4,IWNO) $ DO 130 I=IM,IP $ XK(I)=XW $ YK(I)=YW       A2.74
                  130 CONTINUE                                                         A2.75
                      IF(IWNO.GE.MAXWIND) GO TO 135                                    A2.76
                      IF(IT.GE.IWIND(7,IWNO+1)) GO TO 100                              A2.77
                  135 ITW=IWIND(8,IWNO)                                                A2.78
                  140 CONTINUE                                                         A2.79
                      IF(IT.GE.ITO.AND.MOD(IT,MODOUT).EQ.0)CALL INPOUT(V,LAYER,20,IT,F) A2.80
                      IF(IT.LT.IPO.OR.MOD(IT,IPMOD).NE.0) GO TO 800                    A2.81
                      CALL PRTMAX(T,4HNONE,1H ,28HADVECTED TEMPERATURE ANOMALY,28,IT)  A2.82
                      CALL PRTMAX(CT,4HNONE,1H ,14HCONCENTRATIONS,14,IT)               AD.65
                  800 CONTINUE                                                         AD.66
                  999 CONTINUE $ END                                                   A1.27


 CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

     67     I                  CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
```

```
SUBROUTINE PRTMAX   7600→7600 OPT=2                   FTN 4.6+439/040    09 MAR 77 01.15.36$ BKY PAGE      1

              SUBROUTINE PRTMAX(S,D,IS,NAME1,NC,IT)                                  A1.28
      C       ******** 7600 VERSION *************                                    A1.29
              COMMON/A/ ME,NE,ROT,DL,MAXDIM,MODEL,IDIM,NO,MINX,MINY,MAXX,MAXY,DTA.2
             *,FRACTX,FRACTY,IDIM4,NXP1                                              A.3
             *,NEM,TEM,TE,TWO                                                        COM.1
              COMMON/1/IK,IWW,JK,I,A,B,NOP,MF,ML,MMF,NN,NW,IW(948)                   A1.31
              DIMENSION S(1),D(1)                                                    A1.32
              DATA(LU=3) ,(MAXNAME= 8) ,(NCPW=10),(LEV=1)                            A1.33
              NW=(NC-1)/NCPW+1                                                       A1.34
              NOP=(ME-1)/15+1 $ MF=-14                                               A1.35
              CALL XMIT(NAME1,IW,NW,1,1) $CALL XMIT(1H ,IW(NW+1),MAXNAME-NW,0,1)A1.36
              DO 100 IK=1,NOP $ MF=MF+15 $ ML=MINO(MF+14,ME)                         A1.37
              IF(NE.LT.25.AND.MOD(IK,2).EQ.0.AND.IS.EQ.1H ) GO TO 30                 A1.38
           20 WRITE(LU,1000) (IW(I),I=1,MAXNAME),LEV,IT                              A1.40
      C---------DEBUG----------                                                      A2.83
         1000 FORMAT(1H 20X8A10,I3,7X5HTIME=I10)                                     A2.84
           30 WRITE(LU,1001)(I,I=MF,ML)                                              A1.42
         1001 FORMAT(//6X15I8)                                                       A1.43
              IF(D(1).NE.4HNONE) GO TO 60                                            A2.85
              DO 51 J=1,NE $ KJ=(J-1)*ME $ KK=KJ+ML $ KJ=KJ+MF                       A2.86
      C---------DEBUG----------                                                      A2.87
              DO 49 I=KJ,KK $ IF(S(I).NE.0.)  GO TO 50                               A2.88
           49 CONTINUE $ GO TO 51                                                    A2.89
           50 WRITE(LU,2000)IS,J,(S(JK),JK=KJ,KK)                                    A2.90
           51 CONTINUE                                                               A2.91
      C---------DEBUG----------                                                      A2.92
         2000 FORMAT(A1,I4,3X15E8.3)                                                 A2.93
              GO TO 100                                                              A1.48
           60 DO 90 J=1,NE $ NN=10 $ IJ=(J-1)*ME $ KK=IJ+ML $ KJ=IJ+MF               A2.94
              DO 80 JK=KJ,KK $ NN=NN+1                                               A2.95
              MM=JK-1 $ IF(MM.LE.IJ) MM=JK $ NM=JK-ME $ IF(NM.LE.0)NM=JK             A2.96
              X=(D(JK)+D(MM))*.5 $ Y=(S(JK)+S(NM))*.5                                A1.51
              IF(X.NE.0..OR.Y.NE.0.)GO TO 70 $ IW(NN)=1H  $ GO TO 80                 A1.52
           70 A=SQRT(X*X+Y*Y) $ B=AMOD(ROT-ATAN2(Y,X)*57.295779513,360.)             A1.53
              ENCODE(9,3000,IWW)A,B $ IW(NN)=IWW.OR.770000000000B                    A1.55
         3000 FORMAT(F5.0,F4.0)                                                      A1.56
           80 CONTINUE                                                               A1.57
           90 WRITE(LU,4000)IS,J,(IW(K),K=11,NN)                                     A1.58
         4000 FORMAT(A1,I4,3X15A8)                                                   A1.59
          100 CONTINUE $ RETURN                                                      A1.60
              END                                                                    A1.61
```

```
SUBROUTINE XMIT      7600→7600 OPT=2                  FTN 4.6+439/040    09 MAR 77 01.15.36$ BKY PAGE      1

              SUBROUTINE XMIT(IA,IB,N,JA,JB) $DIMENSION IA(1),IB(1)$I=J=1            A1.62
              DO 5 NN=1,N $ IB(J)=IA(I) $ J=J+JB $ I=I+JA                            A1.63
            5 CONTINUE $ RETURN                                                      A1.64
              END                                                                    A1.65
```

```
SUBROUTINE BDVECT   7600-7600 OPT=2                                    FTN 4.6+439/040    09 MAR 77 01.15.36$ BKY PAGE      1

               SUBROUTINE BDVECT(IQ)                                                      A2.97
      C        MODIFIED TO SINGLE DIMENSION BY R. BAUER COMPASS SYSTEMS, INC. MARAD.104
               COMMON/A/ ME,NE,ROT,DL,MAXDIM,MODEL,IDIM,NO,MINX,MINY,MAXX,MAXY,DTA.2
              *,FRACTX,FRACTY,IDIM4,NXP1                                                  A.3
              *,NEM,TEM,TE,TWO                                                            COM.1
               COMMON CT(300),FXT(300),FYT(300),RXT(300),RYT(300)                         COM.2
              *,T(300),VX(300),VY(300)                                                    COM.3
               COMMON/1/CACC( 90),FXACC( 90),FYACC( 90),RXACC( 90),RYACC( 90),            A3.46
              * F2XACC( 90),F2YACC( 90),C(1)                                              A3.47
               CALL XMIT(.0,CACC,IQ*IDIM4,0,1)                                            A2.98
      C                  MAIN CALCULATIONAL LOOPS FOLLOW. ENDING AT                       AD.115
      C                  STATEMENTS LABELED 500 AND 800                                   A3.14
               DO 800 K=1,NE $ J=2 $ IM=(K-1)*ME $ DO 500 I=1,ME $ IK=I+IM                A3.15
               IF(CT(IK).LT.TE) GO TO 500                                                 A2.100
               IKM=MAXO(IK-1,IM+1)                                                        A3.16
               SIGMAX=((.5+FXT(IK))*VX(IK)+(.5-FXT(IK))*VX(IKM))*FRACTX                   A2.102
               IKM=MAXO(IK-ME,I)                                                          A2.103
               SIGMAY=((.5+FYT(IK))*VY(IK)+(.5-FYT(IK))*VY(IKM))*FRACTY                   A2.104
               SIGMAY=-SIGMAY                                                             A3.17
      C        PLUS V DIRECTION IS NEGATIVE OF Y INDEX DIRECTION                          A3.18
               SIGNX = SIGNY = 1.                                                         AD.121
               IF(SIGMAX.NE.0.) SIGNX=SIGN(1.,SIGMAX)                                     A1.70
               IF(SIGMAY.NE.0.) SIGNY=SIGN(1.,SIGMAY)                                     A1.71
               I1 = I + (SIGNX + 0.1) $ IF(SIGMAX.EQ.0.) I1 = I                           AD.124
               J1 = J + (SIGNY + 0.1) $ IF(SIGMAY.EQ.0.)  J1 = J                          AD.125
               IF(I1.LT.1) I1=NXP1                                                        A2.105
               PIJX=(SIGNX*2.*(FXT(I K)+SIGMAX)+RXT(I K)-1.0)/(2.*RXT(IK))                AD.127
               PIJY=(SIGNY*2.*(FYT(I K)+SIGMAY)+RYT(I K)-1.0)/(2.*RYT(IK))                AD.128
      C                  NORMAL CONDITIONS                                                AD.129
      C                  IF PX,PY<0 OR PX,PY>1 MODIFICATIONS ARE MADE                     AD.130
               NCOUNT=0 $ IF(PIJX.GE.TE.AND.PIJX.LE.TEM) GO TO 150                        A2.106
               NCOUNT=1 $ IF(PIJX.LT.TE) PIJX=0.                                          A2.107
               IF(PIJX.GT.TEM) PIJX=1.                                                    A2.108
           150 IF(PIJY.GE.TE.AND.PIJY.LE.TEM) GO TO 160                                   A2.109
               NCOUNT=1 $ IF(PIJY.LT.TE) PIJY=0.                                          A2.110
               IF(PIJY.GT.TEM) PIJY=1.                                                    A2.111
           160 C1IJT1=CT(I K)*PIJX*(1.0-PIJY)                                             AD.137
               C2IJT1=CT(I K)*(1.0-PIJX)*PIJY                                             AD.138
               C3IJT1=CT(I K)*PIJX*PIJY                                                   AD.139
               C4IJT1=CT(I K)*(1.0-PIJX)*(1.-PIJY)                                        AD.140
               IJ=(J-1)*IDIM+I                                                            AD.141
               I1J=(J-1)*IDIM+I1 $ IJ1=(J1-1)*IDIM+I $ I1J1=(J1-1)*IDIM+I1                AD.142
               CACC(I1 J)=CACC(I1 J)+C1IJT1                                               AD.143
               CACC(I J1)=CACC(I J1)+C2IJT1                                               AD.144
               CACC(I1 J1)=CACC(I1 J1)+C3IJT1                                             AD.145
               CACC(I J)=CACC(I J)+C4IJT1                                                 AD.146
      C        *********    SPECIAL HEAT CALL BYPASS    ***************                   A2.112
               IF(IQ.EQ.1) GO TO 500                                                      A2.113
               F1XT1 = (PIJX * RXT(I K) - 1.0) * 0.5 * SIGNX                              AD.147
               F1YT1 = (1.0 - RYT(I K) * (1.0 - PIJY)) * 0.5 * SIGNY                      AD.148
               F2XT1 = (1.0 - RXT(I K) * (1.0 - PIJX)) * 0.5 * SIGNX                      AD.149
               F2YT1 = (PIJY * RYT(I K) - 1.0) * 0.5 * SIGNY                              AD.150
               FXACC(I1 J)=FXACC(I1 J)+F1XT1*C1IJT1                                       AD.151
               FYACC(I1 J)=FYACC(I1 J)+F1YT1*C1IJT1                                       AD.152
               FXACC(I J1)=FXACC(I J1)+F2XT1*C2IJT1                                       AD.153
               FYACC(I J1)=FYACC(I J1) + F2YT1*C2IJT1                                     AD.154
               FXACC(I1 J1) = FXACC(I1 J1) + F1XT1*C3IJT1                                 AD.155
      FYACC(I1 J1) = FYACC(I1 J1) + F2YT1*C3IJT1                        AD.156
      FXACC(I J) = FXACC(I J) + F2XT1*C4IJT1                            AD.157
      FYACC(I J) = FYACC(I J) + F1YT1*C4IJT1                            AD.158
      F2XACC(I1 J) = F2XACC(I1 J) + F1XT1*F1XT1*C1IJT1                  AD.159
      F2YACC(I1 J) = F2YACC(I1 J) + F1YT1*F1YT1*C1IJT1                  AD.160
      F2XACC(I J1) = F2XACC(I J1) + F2XT1*F2XT1*C2IJT1                  AD.161
      F2YACC(I J1) = F2YACC(I J1) + F2YT1*F2YT1*C2IJT1                  AD.162
      F2XACC(I1 J1) = F2XACC(I1 J1) + F1XT1*F1XT1*C3IJT1                AD.163
      F2YACC(I1 J1) = F2YACC(I1 J1) + F2YT1*F2YT1*C3IJT1                AD.164
      F2XACC(I J) = F2XACC(I J)+ F2XT1*F2XT1*C4IJT1                     AD.165
      F2YACC(I J) = F2YACC(I J)+F1YT1*F1YT1*C4IJT1                      AD.166
      R1XT1 = (PIJX * RXT(I K)) * (PIJX * RXT(I K))                     AD.167
      R1YT1 = ((1.0 - PIJY) * RYT(I K)) * (( 1.0 - PIJY) * RYT(I K))    AD.168
      R2XT1 = ((1.0 - PIJX) * RXT(I K)) * (( 1.0 - PIJX) * RXT(I K))    AD.169
      R2YT1 = (PIJY * RYT(I K)) * (PIJY * RYT(I K))                     AD.170
      RXACC(I1 J)=RXACC(I1 J)+ R1XT1* C1IJT1                            AD.171
      RYACC(I1 J)=RYACC(I1 J)+ R1YT1* C1IJT1                            AD.172
      RXACC(I J1)=RXACC(I J1) + R2XT1* C2IJT1                           AD.173
      RYACC(I J1)=RYACC(I J1)+ R2YT1* C2IJT1                            AD.174
      RXACC(I1 J1)=RXACC(I1 J1)+R1XT1*C3IJT1                            AD.175
      RYACC(I1 J1)=RYACC(I1 J1)+R2YT1*C3IJT1                            AD.176
      RXACC(I J) = RXACC(I J)+ R2XT1*C4IJT1                             AD.177
      RYACC(I J)=RYACC(I J)+R1YT1 *C4IJT1                               AD.178
      IF(NCOUNT.EQ.0) GO TO 500                                         AD.179
      SIGMX = SIGMAX+FXT(I K)                                           AD.180
      SIGMY = SIGMAY + FYT(I K)                                         AD.181
      IF(PIJX.GT.0.0001) GO TO 490                                      AD.182
      FXACC(I1 J) = FXACC(I1 J) - C1IJT1* F1XT1                         AD.183
      F2XACC(I1 J)=F2XACC(I1 J)-C1IJT1* F1XT1*F1XT1                     AD.184
      FXACC(I J1) = FXACC(I J1) - C2IJT1*(F2XT1-SIGMX)                  AD.185
      F2XACC(I J1)=F2XACC(I J1)-C2IJT1*(F2XT1*F2XT1-SIGMX*SIGMX)        AD.186
      FXACC(I1 J1)=FXACC(I1 J1)-C3IJT1* F1XT1                           AD.187
      F2XACC(I1 J1)=F2XACC(I1 J1)-C3IJT1* F1XT1*F1XT1                   AD.188
      FXACC(I J)= FXACC(I J) - C4IJT1*(F2XT1 - SIGMX)                   AD.189
      F2XACC(I J) = F2XACC(I J)-C4IJT1*(F2XT1*F2XT1-SIGMX*SIGMX)        AD.190
      GO TO 492                                                         AD.191
  490 IF(PIJX.LT.0.9999) GO TO 492                                      AD.192
      FXACC(I1 J) = FXACC(I1 J)- C1IJT1*(F1XT1-SIGMX +1.*SIGNX)         AD.193
      F2XACC(I1 J)=F2XACC(I1 J)-C1IJT1*(F1XT1*F1XT1-                    AD.194
     1 (SIGMX - SIGNX) * (SIGMX - SIGNX))                               AD.195
      FXACC(I J1) = FXACC(I J1) - C2IJT1* F2XT1                         AD.196
      F2XACC(I J1)=F2XACC(I J1)-C2IJT1* F2XT1*F2XT1                     AD.197
      FXACC(I1 J1) = FXACC(I1 J1) - C3IJT1*(F1XT1 - SIGMX +1.*SIGNX)    AD.198
      F2XACC(I1 J1) =F2XACC(I1 J1)-C3IJT1*(F1XT1*F1XT1-                 AD.199
     1 (SIGMX - SIGNX) * (SIGMX - SIGNX))                               AD.200
      FXACC(I J) = FXACC(I J) - C4IJT1* F2XT1                           AD.201
      F2XACC(I J)=F2XACC(I J)-C4IJT1* F2XT1*F2XT1                       AD.202
  492 IF(PIJY.GE.0.0001) GO TO 494                                      AD.203
      FYACC(I1 J) = FYACC(I1 J) - C1IJT1*(F1YT1 - SIGMY)                AD.204
      F2YACC(I1 J) = F2YACC(I1 J) - C1IJT1*(F1YT1*F1YT1-SIGMY*SIGMY)    AD.205
      FYACC(I J1) = FYACC(I J1) - C2IJT1* F2YT1                         AD.206
      F2YACC(I J1) = F2YACC(I J1) - C2IJT1* F2YT1*F2YT1                 AD.207
      FYACC(I1 J1) = FYACC(I1 J1) - C3IJT1* F2YT1                       AD.208
      F2YACC(I1 J1) = F2YACC(I1 J1) - C3IJT1* F2YT1* F2YT1              AD.209
      FYACC(I J) = FYACC(I J) - C4IJT1*(F1YT1 - SIGMY)                  AD.210
      F2YACC(I J)=F2YACC(I J)-C4IJT1*(F1YT1*F1YT1-SIGMY*SIGMY)          AD.211
      GO TO 500                                                         AD.212
     494 IF(PIJY.LT.0.9999) GO TO 500                                         AD.213
         FYACC(I1 J) = FYACC(I1 J) - C1IJT1* F1YT1                            AD.214
         F2YACC(I1 J)=F2YACC(I1 J)-C1IJT1*(F1YT1*F1YT1)                       AD.215
         FYACC(I J1) = FYACC(I J1) - C2IJT1*(F2YT1 - SIGMY +1.*SIGNY)         AD.216
         F2YACC(I J1)=F2YACC(I J1)-C2IJT1*(F2YT1*F2YT1-                       AD.217
        1 (SIGMY - SIGNY) * (SIGMY - SIGNY))                                  AD.218
         FYACC(I1 J1)=FYACC(I1 J1)-C3IJT1*(F2YT1-SIGMY +1.*SIGNY)             AD.219
         F2YACC(I1 J1)=F2YACC(I1 J1)-C3IJT1*(F2YT1*F2YT1-                     AD.220
        1 (SIGMY - SIGNY) * (SIGMY - SIGNY))                                  AD.221
         FYACC(I J) = FYACC(I J) - C4IJT1* F1YT1                              AD.222
         F2YACC(I J)=F2YACC(I J)-C4IJT1* F1YT1*F1YT1                          AD.223
     500 CONTINUE                                                             AD.224
         IF(K.LE.1) GO TO 610 $ IL1=(K-2)*ME $ IJ1=0                          A3.19
     510 DO 600 I=1,ME $ IL=IL1+I $ IJ=IJ1+I                                  A3.20
    C                   VALUES AT NEXT TIME STEP ARE CALCULATED               AD.227
         CT(I L) = CACC(I J)                                                  AD.228
    C    **********   SPECIAL HEAT CALL BYPASS   ***************              A2.114
         IF(IQ.EQ.1) GO TO 600                                                A2.115
         CTREPL = CACC(I J)                                                   AD.229
         IF(CTREPL.LT.1.E-10) GO TO 599 $ CTREPL=1./CTREPL                    A1.72
         FX=FXT(I L)=FXACC(I J)*CTREPL                                        A1.73
         FY=FYT(I L)=FYACC(I J)*CTREPL                                        A1.74
         RXMAX=ABS(2.*FX-1.) $ RXMIN=ABS(2.*FX+1.)                            A1.75
         IF(RXMIN.GT.RXMAX) RXMIN=RXMAX                                       AD.237
         RYMAX=ABS(2.*FY-1.) $ RYMIN=ABS(2.*FY+1.)                            A1.76
         IF(RYMIN.GT.RYMAX) RYMIN = RYMAX                                     AD.240
         RXT(IL)=SQRT(RXACC(IJ)*CTREPL+12.*(-FX*FX+F2XACC(IJ)*CTREPL))        A2.116
         RYT(IL)=SQRT(RYACC(IJ)*CTREPL+12.*(-FY*FY+F2YACC(IJ)*CTREPL))        A2.117
         IF(RXT(I L).GT.RXMIN) RXT(I L)=RXMIN                                 AD.245
         IF(RYT(I L).GT.RYMIN) RYT(I L)=RYMIN      $ GO TO 600                AD.246
     599 FXT(IL)=FYT(IL)=RXT(IL)=RYT(IL)=0.                                   AD.247
     600 CONTINUE $ IF(IL1-NO+ME+ME) 610,605,800                              A3.21
     605 IL1=(NE-1)*ME $ IJ1=IDIM $ GO TO 510                                 A3.22
     610 DO 650 I=1,IQ $ IL=(I-1)*IDIM4+IDIM+1                                A3.23
         CALL XMIT(CACC(IL),CACC(IL-IDIM),IDIM+2,1,1)                         A3.24
     650 CALL XMIT(0.,CACC(IL+IDIM),ME+1,0,1)                                 A3.25
     800 CONTINUE $ RETURN $ END                                              A3.26
```

```
SUBROUTINE INPOUT    7600-7600 OPT=2                    FTN 4.6+439/040    09 MAR 77 01.15.36  BKY PAGE     1

                SUBROUTINE INPOUT(A,LAYER,ITY,IT,FACOUT)                        A1.81
                COMMON/A/ ME,NE,ROT,DL,MAXDIM,MODEL,IDIM,NO,MINX,MINY,MAXX,MAXY,DTA.2
               *,FRACTX,FRACTY,IDIM4,NXP1                                       A.3
               *,NEM,TEM,TE,TWO                                                 COM.1
                COMMON/1/IW(960)                                                A1.83
                DIMENSION A(1),IREC(2)                                          A1.84
                DATA(IREC=0,0),(M24=7777 7777B),(M12=7777B),(LEN=500),(J=0)     A1.85
C                   -LAYER=READ,+LAYER=WRITE                                    A1.86
                PRINT 1000,IT,LAYER,ITY,NE,ME,MODEL,IREC                        A1.87
           1000 FORMAT(10H INPOUT      10I10)                                   A1.88
                FAC=FACOUT                                                      A3.27
                ID1=SHIFT(IT,36).OR.SHIFT(IABS(LAYER),12)                       A1.90
                ID2=SHIFT(IABS(ITY),48).OR.SHIFT(NE,36).CR.SHIFT(ME,24).OR.MODEL A2.120
                KP=3 $ IF(ITY.EQ.20) KP=6 $ LU=1 $ IF(KP.EQ.6) LU=2             A3.28
                DO 600 K=1,KP $ KK=(K-1)*MAXDIM+1                               A3.29
                IF(K.EQ.1.AND.ITY.EQ.20) FACOUT=1.E-9                           A3.30
C               VOLUME RANGE ALLOWED- 1.E6 TO 8.E13                             A3.31
                IF(LAYER.GT.0) GO TO 80 $ FACIN=1./FACOUT                       A3.32
                NW=M24 $ IF(K.NE.1.AND.ITY.LT.0) KK=KK-MAXDIM $ GO TO 90        A3.33
             80 NW=2 $ IW(1)=ID1 $ IW(2)=ID2                                    A2.125
             90 DO 500 I=1,NO $ NX=(I-1)/NE $ NY=I-NX*NE $ KKII=(NY-1)*ME+NX+KK  A2.126
                IF(LAYER.GT.0) GO TO 400 $ IF(NW.LT.LEN) GO TO 150              A2.127
                IF(NW.EQ.LEN.AND.J.NE.1) GO TO 150                              A1.95
C                     INPUT                                                     A1.96
            100 BUFFER IN(LU,1)(IW(1),IW(LEN)) $ IREC(LU)=IREC(LU)+1 $ NW=2     A3.34
                IF(UNIT(LU))140,110,120                                         A3.35
           110  IERR=1 $ GO TO 130                                              A1.99
           120  IERR=2                                                          A1.100
            130 PRINT 1000,IW(1),IW(2),ID1,ID2 $ CALL IOERR(IERR,IREC,LU)$GOTO100 A3.36
           140  IF(IW(1).NE.ID1.OR.IW(2).NE.ID2) GO TO 100                      A1.102
           150  J=MOD(I,5)+1 $ GO TO (250,210,220,230,240),J                    A1.103
            210 NW=NW+1 $ IWW=IW(NW) $ N=SHIFT(IWW,-36) $ GO TO 300             A2.129
           220  N=SHIFT(IWW,-12) $ GO TO 260                                    A1.105
           230  N=IWW.AND.M12 $ NW=NW+1 $ IWW=IW(NW)                            A1.106
                N=SHIFT(N,12).OR.SHIFT(IWW,12).AND.M12 $ GO TO 270              A1.107
           240  N=SHIFT(IWW,-24)     $ GO TO 260                                A1.108
           250  N=IWW                                                           A1.109
           260  N=N.AND.M24                                                     A1.110
           270  IF(N.GT.4000 0000B) N=N.OR ..NOT. M24                           A1.111
            300 A(KKII)=FLOAT(N)*FACIN $ GO TO 500                              A2.130
C                     OUTPUT                                                    A1.114
            400 CONTINUE                                                        A2.131
                N=A(KKII)*FACOUT $ N=N.AND.M24                                  A1.117
                J=MOD(I,5)+1 $ GO TO (450,410,420,430,440),J                    A1.118
           410  NW=NW+1 $ IW(NW)=SHIFT(N,36) $ GO TO 470                        A1.119
           420  IW(NW)=IW(NW).OR.SHIFT(N,12) $ GO TO 470                        A1.120
           430  IW(NW)=IW(NW).OR.SHIFT(N,-12) $ NW=NW+1  $ N=N.AND.M12          A1.121
                IW(NW)=SHIFT(N,48) $ GO TO 470                                  A1.122
           440  IW(NW)=IW(NW).OR.SHIFT(N,24) $ GO TO 470                        A1.123
           450  IW(NW)=IW(NW).OR.N                                              A1.124
            470 IF(I.EQ.NO) GO TO 475    $ IF(NW.LT.LEN) GO TO 500              A1.125
                IF(NW.EQ.LEN.AND.J.NE.1) GO TO 500                              A1.126
           475  BUFFER OUT(2,1)(IW(1),IW(NW)) $ IREC(2)=IREC(2)+1 $ NW=2        A1.127
                IF(UNIT(2))500,480,490                                          A1.128
           480  IERR=3 $ GO TO 495                                              A1.129
           490  IERR=4                                                          A1.130
            495 CALL IOERR(IERR,IREC,LU) $ GO TO 475                            A3.37
```

```
      SUBROUTINE INPOUT    7600→7600 OPT=2                       FTN 4.6+439/040     09 MAR 77 01.15.36# BKY PAGE      2

             500  CONTINUE                                                A1.132
              600 FACOUT=FAC $ RETURN $ END                               A3.38
```

```
      SUBROUTINE IOERR     7600→7600 OPT=2                       FTN 4.6+439/040     09 MAR 77 01.15.36# BKY PAGE      1

                      SUBROUTINE IOERR(L,IREC,LU) $ DIMENSION LAB(4),IREC(1)          A3.39
                      DATA(LAB=10HEOF ERR IN,10HPAR ERR IN,10HEOT ERROUT,10HPAR ERROUT) A1.137
                     *,(K=0)                                                          A3.40
                      PRINT 1000,LAB(L),IREC(LU),LU $ IF(L-2) 4,3,2                    A3.41
                 1000 FORMAT(A11,2I10)                                                A3.42
                    2 BACKSPACE LU $ ENDFILE LU $ BACKSPACE LU                        A3.43
                      IF(L.EQ.3) GO TO 4 $ ENDFILE LU $ BACKSPACE LU                   A3.44
                    3 IREC(LU)=IREC(LU)-1 $ K=K+1 $ IF(K.LT.20) RETURN                A3.45
                    4 STOP                                                            A1.143
                        END                                                           A1.144
```

```
SUBROUTINE THERMAL  7600-7600 OPT=2                    FTN 4.6+439/040    09 MAR 77 01.15.360 BKY PAGE      1

                   SUBROUTINE THERMAL                                                  T1.9
                   COMMON/A/ ME,NE,ROT,DL,MAXDIM,MODEL,IDIM,NO,MINX,MINY,MAXX,MAXY,DTA.2
                  *,FRACTX,FRACTY,IDIM4,NXP1                                           A.3
                  *,NEM,TEM,TE,TWO                                                     COM.1
                   COMMON CT(300),FXT(300),FYT(300),RXT(300),RYT(300)                  COM.2
                  *,T(300),VX(300),VY(300)                                             COM.3
                   COMMON/ABC/TA(300),EW(300),EA(300),XK(300),VK(300)                  COM.5
                  * ,QB(300),QH(300),QE(300)                                           COM.6
                   COMMON/THERM/IT,ITS,IPMOD,TL,CC,UU,K,C,EC,EK,KI,EKI,Q2T,DD,R        COM.7
                  *,IPO                                                                T3.28
                   REAL K,KI                                                           THERM.6
                   COMMON/1/IW(960)                                                    T2.5
                   DIMENSION TW(1),V(1) $ EQUIVALENCE (T,TW),(CT,V)                    T3.1
                   DATA (EC=.4),(EK=.5),(C=.1),(K=.35)                                 T2.7
                  *,(RH=72.5),(CC=3.2),(J=0),(QSR=160.)                                T2.8
C                  DEFAULT RELATIVE HUMIDITY = 75 0/0 // CLOUD COVER = 3.2 SKY/10      T3.2
C                  QSR SOLAR HEATING EFFECT DEFAULT CONSTANT THRU 24 HOUR CYCLE        T3.3
C                  INPUT T(I) IS TEMPERATURE ANOMALY OF HEATED FLUID IN THE CELL       T3.4
C                  DEFAULT AMBIENT WATER TEMPERATURE - TWO = 10 C DEGREES              T3.5
C                WATER                                                                 TH.137
C                  COMPUTES THE WATER VAPOR PRESSURE OF AIR                            TH.138
C                  COMPUTES THE SATURATED WATER VAPOR PRESSURE AT SEA SFC TEMP         TH.139
C                  COMPUTES THE NEW TA ARRAY                                           TH.140
                   DO 101 N=1,NE $ IP=N*ME $ IM=IP-ME+1                                T2.10
                   DO 100 I=IM,IP $ MM=MAXO(I-1,IM) $ MP=MINO(I+1,IP)                  T2.11
                   NM=I-ME $ IF(N.EQ.1) NM=I $ NP=I+ME $ IF(N.EQ.NE) NP=I              T2.12
                   IF(CT(I)+CT(NP)+CT(MP)+CT(MM)+CT(NM).LE.TE*5.) GO TO 99             T3.6
C                  INTERPOLATE HEATED FLUID TEMP ANOMALY TO CELL TEMPERATURE           T3.7
                   TW(I)=T(I)*RXT(I)*RYT(I)+TWO $ TA(I)=TWO-2.                         T3.8
C                     AIR TEMP IS RESET AT EACH STEP TO AMBIENT TW -2C DEGREES         T3.9
C                  TO SIMULATE WIND MIXING EFFECT                                      T3.10
                   AK=373.16/(TW(I)+273.16)                                            T3.11
                   AK1=(1.-AK)*7.90298 $ AK2=ALOG10(AK)*5.02808                        T1.16
                   AK3=1.-10.**((1.-1./AK)*11.344) $ AK4=10.**(-3.49149*AK-1.)-1.      T1.17
                   EW(I)=10.**(AK1+AK2+AK3*1.3816E-7+AK4*8.1328E-3)*1013.246           T1.18
                   IF(EA(I).EQ.0.) EA(I)=.98*EW(I)                                     T2.15
                   GO TO 100                                                           T3.12
                99 EW(I)=EA(I)=TA(I)=0.                                                T3.13
               100 CONTINUE                                                            T2.16
               101 CONTINUE                                                            T2.17
                   CALL XMIT(EW(ME-1),EW(ME),NE,ME,ME)                                 T3.19
                   TT=IT/3600. $ TTT=EXP(-EK*TT) $ TT=EXP(-K*TT)                       T2.18
                   T3=(1.-TTT)*EKI $ T2=(1.-TT)*KI                                     T2.19
                   DO 301 N=1,NEM $ IP=N*ME-1 $ IM=IP-ME+2                             T2.20
                   DO 300 I=IM,IP $ IF(TW(I).LE.TWO+TE) GO TO 300                      T3.15
                   J=I $ IF(N.EQ.1) J=J+ME $ IF(I.EQ.IM) J=J+1                         T2.22
                   JJ=J-1 $ IF(XK(J).LT.0.) JJ=J+1                                     T2.23
                   JI=J+ME $ IF(XK(J).LT.0.) JI=J-ME                                   T2.24
                   DIF =((EW(JJ)-EW(J))*XK(J)+(EW(JI)-EW(J))*VK(J))*TL                 T2.25
                   EWEA=(EW(J)-EA(J))*TTT+(EC+DIF)*T3                                  T2.26
                   EA(I)=EW(J)-EWEA                                                    T2.27
C---------BYPASS FOR CONSTANT AIR TEMPERATURE                                          T2.28
                   GO TO 300                                                           T2.29
                   DIF=((TW(JJ)-TW(J))*XK(J)+(TW(JI)-TW(J))*VK(J))*TL                  T3.16
                   TWTA=(TW(J)-TA(J))*TT+(C+DIF)*T2                                    T3.17
                   TA(I)=TW(J)-TWTA                                                    T3.18
               300 CONTINUE                                                            T2.36
```

```
                301 CONTINUE                                                          T2.37
                    CALL XMIT(EA(NO-1),EA(NO),NE,-ME,-ME)                             T2.38
  60                CALL XMIT(EA(NO-ME),EA(NO),ME,-1,-1)                              T2.39
              C     CALL XMIT(TA(NO-1),TA(NO),NE,-ME,-ME)                             T2.40
              C     CALL XMIT(TA(NO-ME),TA(NO),ME,-1,-1)                              T2.41
              C   CALC                                                                TH.278
              C     COMPUTES THE EFFECTIVE BACK RADIATION                             TH.279
  65          C     COMPUTES THE CONVECTIVE TRANSFER OF SENSIBLE HEAT                 TH.280
              C     COMPUTES THE TRANSFER OF LATENT HEAT BY EVAPORATION               TH.281
                    DO 401 N=1,NE $ IP=N*ME $ IM=IP-ME+1                              T2.42
                    DO 400 I=IM,IP $ L=I                                              T2.43
                    IF(V(I).GT.TE) GO TO 350                                          T3.19
  70                TA(I)=QB(I)=QH(I)=QE(I)=TW(I)=0. $ GO TO 400                      T3.20
              C     UNITS-----QB,QH,QE(QSR) ARE IN GM-CAL/CM**2/DAY                   T3.21
                350 QB(I)=(297.-1.86*TW(I)-.95*RH)*(1.-.0765*CC)                      T3.22
                    SS=SQRT(XK(I)*XK(I)+YK(I)*YK(I))*.077+.26                         T2.48
                    QH(I)=39.*SS*(TW(I)-TA(I))                                        T3.23
  75                QE(L)=53.9*SS*(EW(I)*.98-EA(I))                                   T2.50
                    TWTA=(QB(L)+QH(L)+QE(L)-QSR)*Q2T/CT(I)                            T2.51
                    TW(I)=TW(I)-TWTA                                                  T3.24
              C     OUTPUT T(I) IS THE CELL TEMPERATUREFOR AFFECTED CELLS             T3.25
              C     ADJUSTED FOR SOLAR NET RADIATION (QSR) AND OTHER EFFECTS.         T3.26
  80            400 CONTINUE                                                          T2.53
                401 CONTINUE                                                          T2.54
                    IF(IT.LT.IPO.OR.MOD(IT,IPMOD).NE.0) GO TO 500                     T3.27
                    CALL PRTMAX(EA,4HNONE,1H ,32HWATER VAPOR PRESSURE OF AIR (MB),32, T1.47
                   *IT)                                                               T1.48
  85                CALL PRTMAX(EW,4HNONE,1H ,41HEW-SATURATED WATER V. P. AT SEA TEMP T1.49
                   *(MB),41,IT)                                                       T1.50
                    CALL PRTMAX(TA,4HNONE,1H ,15HAIR TEMPERATURE,15,IT)               T1.51
                    CALL PRTMAX(TW,4HNONE,1H ,20HTW-WATER TEMPERATURE,20,IT)          T1.52
                    IF(L.EQ.1) GO TO 500                                              T2.56
  90                CALL PRTMAX(QB,4HNONE,1H ,47HQB-EFFECTIVE BACK RADIATION ADJ. FOR T1.54
                   *CLOUDINESS,47,IT)                                                 T1.55
                    CALL PRTMAX(QH,4HNONE,1H ,39HQH-CONVECTIVE TRANSFER BY SENSIBLE HET1.56
                   *AT,39,IT)                                                         T1.57
                    CALL PRTMAX(QE,4HNONE,1H ,41HQE-TRANSFER OF LATENT HEAT BY EVAPORAT2.57
  95               *TION,41,IT)                                                       T1.59
                500 CONTINUE                                                          T1.60
                    RETURN                                                            TH.323
                    END                                                               TH.324


CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

     54    I                THERE IS NO PATH TO THIS STATEMENT.
```

```
LOAD MAP.                                          LINK - BKY 6000/7000 8.4      09 MAR 77 01.15.41      PAGE    1

FL REQUIRED TO LOAD    43300
FL REQUIRED TO RUN     35200
INITIAL TRANSFER TO ADVECT      -     13020


BLOCK ASSIGNMENTS.

BLOCK           ADDRESS    LENGTH   FILE

 /A/                100        25
 /1/                125      1700
 /ABC/             2025      4540
 /THERM/           6565        20
ADVECT             6605      5624   LGO
PRTMAX            14431       404   LGO
XMIT              15035        20   LGO
BDVECT            15055      1005   LGO
INPOUT            16062       433   LGO
IOERR             16515        65   LGO
THERMAL           16602       622   LGO
 /STP.END/        17424         1
 /FCL.C./         17425        21
 /QLINLMT/        17446         1
 /QJOBSFL/        17447         2
 /QASAFLG/        17451         1
 /.QCOMP./        17452         1
 /Q8.IO./         17453       244
FTN4LIB           17717         0   FTN4LIB
ALOG              17717        77   FTN4LIB
ATAN2             20016       100   FTN4LIB
BACKSP=           20116        62   FTN4LIB
BUFIN=            20200        61   FTN4LIB
BUFIO=            20261       144   FTN4LIB
BUFOUT=           20425        55   FTN4LIB
CIO=              20502        13   FTN4LIB
COMIO=            20515        61   FTN4LIB
DECODE=           20576       115   FTN4LIB
ENCODE=           20713       144   FTN4LIB
ENDFIL=           21057        70   FTN4LIB
ERM$$             21147      1466   FTN4LIB
EXP               22635       101   FTN4LIB
FECMSK=           22736        41   FTN4LIB
FLTIN=            22777       156   FTN4LIB
FLTOUT=           23155       315   FTN4LIB
FMTAP=            23472       372   FTN4LIB
FORSYS=           24064       761   FTN4LIB
 /QOVERL7/        25045         1
FORUTL=           25046        16   FTN4LIB
GETFIT=           25064        43   FTN4LIB
GOTOER=           25127        14   FTN4LIB
INCOM=            25143       262   FTN4LIB
INPC=             25425       246   FTN4LIB
KODER=            25673       467   FTN4LIB
```

```
LOAD MAP.                                     LINK - BKY 6000/7000 8.4      09 MAR 77 01.15.41      PAGE    2
BLOCK ASSIGNMENTS.

BLOCK          ADDRESS    LENGTH   FILE

KRAKER=          26362       435   FTN4LIB
OUTCOM=          27017       204   FTN4LIB
OUTC=            27223       235   FTN4LIB
RDL=             27460        37   FTN4LIB
RDO=             27517        23   FTN4LIB
RDW=             27542       107   FTN4LIB
SINCOS=          27651        74   FTN4LIB
SQRT             27745        46   FTN4LIB
SYSAID=          30013         1   FTN4LIB
SYS=             30014        34   FTN4LIB
SYS=AID          30050         7   FTN4LIB
SYS=1ST          30057        62   FTN4LIB
UNIT             30141        65   FTN4LIB
UZERO..          30226         2   FTN4LIB
WTH=             30230        30   FTN4LIB
WTL=             30260        27   FTN4LIB
WTW=             30307        66   FTN4LIB
XTOY=            30375         7   FTN4LIB
 //              30404      4540
```

```
NAME TEST BOX MODEL
GRID  5  533333.3333       0.    3.2E-6       42.      .003
LAYS    1      77777                     .99          1.022
TIME   600       1200      1200      2400       600      1200       1200
SORC  3  3  3  3 1 1       1200   7777777     4.76E8   4.30E7
COMP     99
                    ADVECTED EXCESS HEAT                                                1       TIME=      1200


            1       2       3       4       5
   3   .0       .0      .286E+12.0       .0
INPOUT          1200        1       20        5        5        99       0       0
                    ADVECTED TEMPERATURE ANOMALY                                        1       TIME=      1200


            1       2       3       4       5
   3   .0       .0      .113E+02.0       .0
                    CONCENTRATIONS                                                      1       TIME=      1200


            1       2       3       4       5
   3   .0       .0      .258E+11.0       .0
INPOUT          1800       -1       -2        5        5        99       0       6
INPOUT          1800        1       20        5        5        99      15       6
INPOUT          2400       -1       -2        5        5        99      15      12
                    VELOCITY                                                            1       TIME=      2400


            1       2       3       4       5
   1     10;184  11;183  11;182   8;181
   2     10;160  11;164  10;167   8;157   5; 92
   3     10;138  10;142  10;145   8;142   4; 97
   4     10;125  10;126   9;130   6;128   3;101
   5     10;120  10;120   8;125   5;120   3;101
                    WATER VAPOR PRESSURE OF AIR (MB)                                    1       TIME=      2400


            1       2       3       4       5
   2   .0       .0      .120E+02.120E+02.120E+02
   3   .0       .120E+02.242E+02.121E+02.121E+02
   4   .0       .120E+02.120E+02.119E+02.119E+02
   5   .0       .120E+02.120E+02.119E+02.119E+02
                    EW-SATURATED WATER V. P. AT SEA TEMP (MB)                           1       TIME=      2400


            1       2       3       4       5
   2   .0       .0      .123E+02.123E+02.123E+02
   3   .0       .123E+02.238E+02.132E+02.132E+02
   4   .0       .123E+02.129E+02.123E+02.123E+02
   5   .0       .0      .123E+02.123E+02.123E+02
                    AIR TEMPERATURE                                                     1       TIME=      2400


            1       2       3       4       5
   3   .0       .0      .800E+01.800E+01.0
   4   .0       .0      .800E+01.800E+01.0
                    TW-WATER TEMPERATURE                                                1       TIME=      2400


            1       2       3       4       5
```

B Uc

```
   3   .0       .0      .203E+02.110E+02.0
   4   .0       .0      .107E+02.934E+01.0
                      QB-EFFECTIVE BACK RADIATION ADJ. FOR CLOUDINESS                              1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      .144E+03.157E+03.0
   4   .0       .0      .157E+03.158E+03.0
                      QH-CONVECTIVE TRANSFER BY SENSIBLE HEAT                                      1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      .125E+03.314E+02.0
   4   .0       .0      .286E+02.213E+02.0
                      QE-TRANSFER OF LATENT HEAT BY EVAPORATION                                    1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      ********.113E+02.0
   4   .0       .0      .897E+01.220E+01.0
                      ADVECTED EXCESS HEAT                                                         1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      .677E+12.692E+11.0
   4   .0       .0      .856E+11.137E+11.0
 INPOUT         2400            1        20           5          5         99         18         12
                      ADVECTED TEMPERATURE ANOMALY                                                 1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      .113E+02.108E+02.0
   4   .0       .0      .109E+02.906E+01.0
                      CONCENTRATIONS                                                               1      TIME=     2400

              1        2        3        4        5
   3   .0       .0      .613E+11.652E+10.0
   4   .0       .0      .805E+10.154E+10.0
```